## Description

Initializes a new instance of the **System.Web.HttpUnhandledException** class with the specified error message and inner exception. The message displayed to the client when the exception is thrown. The **System.Exception.InnerException**, if any, that threw the current exception.

HttpUnhandledException

Example Syntax:

**ToString** 

[C#] public HttpUnhandledException(string message, string postMessage, Exception innerException);

[C++] public: HttpUnhandledException(String\* message, String\* postMessage, Exception\* innerException);

[VB] Public Sub New(ByVal message As String, ByVal postMessage As String, ByVal innerException As Exception)

[JScript] public function HttpUnhandledException(message : String, postMessage : String, innerException : Exception);

# Description

Initializes a new instance of the **HttpUnhandledException** class with the specified error messages and inner exception. The message displayed to the client when the exception is thrown. An additional message displayed to the client when the exception is thrown. The **System.Exception.InnerException**, if any, that threw the current exception.

```
ErrorCode
          HelpLink
          HResult
          InnerException
          Message
          Source
          StackTrace
          TargetSite
          HttpUtility class (System.Web)
9
          ToString
10
11
12
    Description
13
          Provides methods for encoding and decoding URLs when processing Web
14
    requests.
15
          HttpUtility
16
          Example Syntax:
17
           ToString
18
19
                                                                        HttpUtility();
                                     public
    [C#]
20
                                                                        HttpUtility();
                                     public:
    [C++]
                                                                               New()
    [VB]
                             Public
                                                       Sub
22
    [JScript] public function HttpUtility();
23
           HtmlAttributeEncode
24
25
```

4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |

19

20

21

22

23

24

25

2

3

[C#] public static string HtmlAttributeEncode(string s); String\* HtmlAttributeEncode(String\* s); [C++]public: static [VB] Public Shared Function HtmlAttributeEncode(ByVal s As String) As String [JScript] public static function HtmlAttributeEncode(s : String) : HTML-encoded string. Minimally string into an converts a

### Description

Minimally converts a string to an HTML-encoded string.

Return Value: The encoded string.

HtmlAttributeEncode converts only quotes (") and ampersands (&) to equivalent character entities. It is considerably faster than the System.Web.HttpUtility.HtmlEncode(System.String) methods. The string to encode.

#### HtmlAttributeEncode

[C#] public static void HtmlAttributeEncode(string s, TextWriter output);
[C++] public: static void HtmlAttributeEncode(String\* s, TextWriter\* output);
[VB] Public Shared Sub HtmlAttributeEncode(ByVal s As String, ByVal output
As TextWriter)

[JScript] public static function HtmlAttributeEncode(s : String, output : TextWriter);

198

## Description

lee@hayes pilc 509-324-9256

Minimally converts a string into an HTML-encoded string and sends the encoded string to a **System.IO.TextWriter** output stream.

HtmlAttributeEncode converts only quotes (") and ampersands (&) to equivalent character entities. It is considerably faster than the System.Web.HttpUtility.HtmlEncode(System.String) methods. The string to encode A System.IO.TextWriter output stream.

HtmlDecode

[C#] public static string HtmlDecode(string s);
[C++] public: static String\* HtmlDecode(String\* s);
[VB] Public Shared Function HtmlDecode(ByVal s As String) As String
[JScript] public static function HtmlDecode(s: String): String; Converts a string that has been HTML-encoded for HTTP transmission into a decoded string.

## Description

Converts a string that has been HTML-encoded for HTTP transmission into decoded string.

Return Value: The decoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents; HTML decoding reverses the encoding. For example, when embedded in a block of text, the characters < and >, are encoded as < and > for HTTP transmission. The string to decode.

HtmlDecode

[C#] public static void HtmlDecode(string s, TextWriter output);
[C++] public: static void HtmlDecode(String\* s, TextWriter\* output);
[VB] Public Shared Sub HtmlDecode(ByVal s As String, ByVal output As TextWriter)

[JScript] public static function HtmlDecode(s : String, output : TextWriter);

## Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Converts a string that has been HTML-encoded into a decoded string, and sends the decoded string to a **System.IO.TextWriter** output stream.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents; HTML decoding reverses the encoding. For example, when embedded in a block of text, the characters < and >, are encoded as < and > for HTTP transmission. The string to decode. A **System.IO.TextWriter** stream of output.

### HtmlEncode

HtmlEncode(string s); public static string [C#] [C++]public: static String\* HtmlEncode(String\* s); [VB] Public Shared Function HtmlEncode(ByVal s As String) As String [JScript] public static function HtmlEncode(s : String) : String; Converts a string into an HTML-encoded string for reliable HTTP transmission from the Web client. server to a

lee⊗haves olic 509-324-9256 200 MSI-863US.APP

## Description

Converts a string to an HTML-encoded string for reliable HTTP transmission from the Web server to a client.

Return Value: The encoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents; HTML decoding reverses the encoding. For example, when embedded in a block of text, the characters < and >, are encoded as < and > for HTTP transmission. The string to encode.

#### HtmlEncode

[C#] public static void HtmlEncode(string s, TextWriter output);
[C++] public: static void HtmlEncode(String\* s, TextWriter\* output);
[VB] Public Shared Sub HtmlEncode(ByVal s As String, ByVal output As TextWriter)

[JScript] public static function HtmlEncode(s : String, output : TextWriter);

## Description

Converts a string into an HTML-encoded string, and returns the output as a TextWriter stream of output.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. HTML encoding converts characters that are not allowed in HTML into character-entity equivalents; HTML

decoding reverses the encoding. For example, when embedded in a block of text, the characters < and >, are encoded as < and > for HTTP transmission. The string to encode A System.IO.TextWriter output stream.

UrlDecode

[C#] public static string UrlDecode(string str);
[C++] public: static String\* UrlDecode(String\* str);
[VB] Public Shared Function UrlDecode(ByVal str As String) As String
[JScript] public static function UrlDecode(str: String): String; Converts a string that has been encoded for transmission in a URL into a decoded string.

## Description

Converts a string that has been encoded for transmission in a URL into a decoded string.

Return Value: The decoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to decode.

#### UrlDecode

[C#] public static string UrlDecode(byte[] bytes, Encoding e); [C++] public: static String\* UrlDecode(unsigned char bytes \_\_gc[], Encoding\* e);

[VB] Public Shared Function UrlDecode(ByVal bytes() As Byte, ByVal e As Encoding)

As String

[JScript] public static function UrlDecode(bytes : Byte[], e : Encoding) : String;

Description

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Converts a URL-encoded byte array into a decoded string, using the specified decoding method.

Return Value: The decoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The byte array to decode. The **System.Text.Encoding** that specifies the decoding method.

UrlDecode

UrlDecode(string Encoding [C#] public static string str, e); Encoding\* UrlDecode(String\* String\* [C++]public: static str. [VB] Public Shared Function UrlDecode(ByVal str As String, ByVal e As String Encoding) As [JScript] public static function UrlDecode(str : String, e : Encoding) : String;

Description

lee@hayes\_plic 509-324-9256 203 MS1-863US.APP

Converts a URL-encoded string into a decoded string, using the specified decoding method.

Return Value: The decoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to decode. The **System.Text.Encoding** that specifies the decoding method.

UrlDecode

[C#] public static string UrlDecode(byte[] bytes, int offset, int count, Encoding e);
[C++] public: static String\* UrlDecode(unsigned char bytes \_\_gc[], int offset, int count, Encoding\* e);
[VB] Public Shared Function UrlDecode(ByVal bytes() As Byte, ByVal offset As Integer, ByVal count As Integer, ByVal e As Encoding) As String
[JScript] public static function UrlDecode(bytes: Byte[], offset: int, count: int, e: Encoding) : String;

## Description

Converts a URL-encoded byte array into a decoded string, using the specified decoding method, starting at the specified position in the array, and continuing for the specified number of bytes.

Return Value: The decoded string.

lee@hayes pilc 509-324-9256 204 *MS1-863US.APP* 

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The array of bytes to decode. The position in the byte to begin decoding. The number of bytes to decode starting at *offset*. The **System.Text.Encoding** object that specifies the decoding method.

## UrlDecodeToBytes

[C#] public static byte[] UrlDecodeToBytes(byte[] bytes);
[C++] public: static unsigned char UrlDecodeToBytes(unsigned char bytes \_\_gc[])
\_\_gc[];

[VB] Public Shared Function UrlDecodeToBytes(ByVal bytes() As Byte) As Byte()

[JScript] public static function UrlDecodeToBytes(bytes : Byte[]) : Byte[];

## Description

Converts a URL-encoded arrray of bytes into a decoded array of bytes.

Return Value: The decoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to

be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The array of bytes to decode.

## UrlDecodeToBytes

[C#] public static byte[] UrlDecodeToBytes(string str);
[C++] public: static unsigned char UrlDecodeToBytes(String\* str) \_\_gc[];
[VB] Public Shared Function UrlDecodeToBytes(ByVal str As String) As Byte()
[JScript] public static function UrlDecodeToBytes(str: String): Byte[]; Converts a
URL-encoded string or byte array into a decoded array of bytes.

## Description

Converts a URL-encoded string into a decoded array of bytes .

\*Return Value: The decoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to decode.

# UrlDecodeToBytes

[C#] public static byte[] UrlDecodeToBytes(string str, Encoding e);
[C++] public: static unsigned char UrlDecodeToBytes(String\* str, Encoding\* e)
\_\_gc[];

[VB] Public Shared Function UrlDecodeToBytes(ByVal str As String, ByVal e As

Encoding)

As

Byte()

[JScript] public static function UrlDecodeToBytes(str : String, e : Encoding) :

Byte[];

Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Converts a URL-encoded string into a decoded array of bytes, using the specified decoding method.

Return Value: The decoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to decode. The **System.Text.Encoding** object that specifies the decoding method.

UrlDecodeToBytes

[C#] public static byte[] UrlDecodeToBytes(byte[] bytes, int offset, int count); [C++] public: static unsigned char UrlDecodeToBytes(unsigned char bytes \_\_gc[], offset, int count) \_gc[]; int [VB] Public Shared Function UrlDecodeToBytes(ByVal bytes() As Byte, ByVal Integer) Byte() offset ByVal count As As As Integer, [JScript] public static function UrlDecodeToBytes(bytes: Byte[], offset: int, int) Byte[]; count

## Description

1

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Converts a URL-encoded arrray of bytes into a decoded array of bytes, starting at the specified position in the array and continuing for the specified number of bytes.

Return Value: The decoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The array of bytes to decode. The position in the byte array to begin decoding. The number of bytes to decode, starting at *offset*.

#### UrlEncode

UrlEncode(byte[] bytes); public static string [C#] static String\* UrlEncode(unsigned char gc[]); [C++]public: [VB] Public Shared Function UrlEncode(ByVal bytes() As Byte) As String String; UrlEncode(bytes: Byte[]) function public static [JScript]

## Description

Converts a byte array into an encoded URL string for reliable HTTP transmission from the Web server to a client.

Return Value: The encoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The byte array to encode.

UrlEncode

[C#] public static string UrlEncode(string str);
[C++] public: static String\* UrlEncode(String\* str);
[VB] Public Shared Function UrlEncode(ByVal str As String) As String
[JScript] public static function UrlEncode(str: String): String; Encodes a URL string for reliable HTTP transmission from the Web server to a client.

## Description

Encodes a URL string for reliable HTTP transmission from the Web server to a client.

Return Value: The encoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The text to encode.

UrlEncode

Encoding UrlEncode(string e); public static string str, [C#] Encoding\* public: static String\* UrlEncode(String\* str, e); [C++][VB] Public Shared Function UrlEncode(ByVal str As String, ByVal e As String As Encoding) [JScript] public static function UrlEncode(str : String, e : Encoding) : String;

## Description

1

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Encodes a URL string for reliable HTTP transmission from the Web server to a client, using the specified encoding method.

Return Value: The encoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The text to encode. The **System.Text.Encoding** object that specifies the encoding scheme.

#### UrlEncode

[C#] public static string UrlEncode(byte[] bytes, int offset, int count);
[C++] public: static String\* UrlEncode(unsigned char bytes \_\_gc[], int offset, int count);
[VB] Public Shared Function UrlEncode(ByVal bytes() As Byte, ByVal offset As Integer, ByVal count As Integer) As String

[JScript] public static function UrlEncode(bytes : Byte[], offset : int, count : int) : String;

## Description

Converts a byte array into a URL-encoded string for reliable HTTP transmission from the Web server to a client, starting at the specified position in the array and continuing for the specified number of bytes. *Return Value:* The encoded string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The byte array to encode. The position in the byte array to begin encoding. The number of bytes to encode, starting at *offset*.

## UrlEncodeToBytes

[C#] public static byte[] UrlEncodeToBytes(byte[] bytes);
[C++] public: static unsigned char UrlEncodeToBytes(unsigned char bytes \_\_gc[])
\_\_gc[];

[VB] Public Shared Function UrlEncodeToBytes(ByVal bytes() As Byte) As Byte()

[JScript] public static function UrlEncodeToBytes(bytes : Byte[]) : Byte[];

# Description

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Converts an array of bytes into a URL-encoded array of bytes.

Return Value: The encoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The array of bytes to encode.

## UrlEncodeToBytes

UrlEncodeToBytes(string str); static byte[] [C#] public [C++] public: static unsigned char UrlEncodeToBytes(String\* str) \_\_gc[]; [VB] Public Shared Function UrlEncodeToBytes(ByVal str As String) As Byte() [JScript] public static function UrlEncodeToBytes(str : String) : Byte[]; Converts a string or a byte array into an encoded array of bytes for reliable HTTP client. Web to a transmission from the server

## Description

Converts a string into a URL-encoded array of bytes.

Return Value: The encoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to

lee@hayes pik 509-324-9256 212 MS1-863US.APP

be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to encode.

UrlEncodeToBytes

[C#] public static byte[] UrlEncodeToBytes(string str, Encoding e);
[C++] public: static unsigned char UrlEncodeToBytes(String\* str, Encoding\* e)
 gc[];

[VB] Public Shared Function UrlEncodeToBytes(ByVal str As String, ByVal e As Encoding)

As Byte()

[JScript] public static function UrlEncodeToBytes(str : String, e : Encoding) : Byte[];

## Description

Converts a string into a URL-encoded array of bytes, using the specified encoding method.

Return Value: The encoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to encode The **System.Text.Encoding** that specifies the encoding method.

UrlEncodeToBytes

[C#] public static byte[] UrlEncodeToBytes(byte[] bytes, int offset, int count);

[C++] public: static unsigned char UrlEncodeToBytes(unsigned char bytes \_\_gc[], gc[]; count) int offset, int [VB] Public Shared Function UrlEncodeToBytes(ByVal bytes() As Byte, ByVal As Byte() Integer) As ByVal count offset As Integer, [JScript] public static function UrlEncodeToBytes(bytes : Byte[], offset : int, count Byte[]; int)

## Description

2

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Converts an array of bytes into a URL-encoded array of bytes, starting at the specified position in the array and continuing for the specified number of bytes.

Return Value: The encoded array of bytes.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not allowed in a URL into character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The array of bytes to encode. The position in the byte array to begin encoding. The number of bytes to encode, starting at *offset*.

#### UrlEncodeUnicode

[C#] public static string UrlEncodeUnicode(string str);
[C++] public: static String\* UrlEncodeUnicode(String\* str);
[VB] Public Shared Function UrlEncodeUnicode(ByVal str As String) As String
[JScript] public static function UrlEncodeUnicode(str : String) : String;

## Description

Converts a string into a Unicode string.

Return Value: The Unicode string.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not legal in a URL to character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to convert.

## UrlEncodeUnicodeToBytes

[C#] public static byte[] UrlEncodeUnicodeToBytes(string str);
[C++] public: static unsigned char UrlEncodeUnicodeToBytes(String\* str) \_\_gc[];
[VB] Public Shared Function UrlEncodeUnicodeToBytes(ByVal str As String) As
Byte()

[JScript] public static function UrlEncodeUnicodeToBytes(str : String) : Byte[];

## Description

Converts a string into a Unicode array of bytes.

Return Value: The Unicode byte array.

If characters such as blanks and punctuation are passed in an HTTP stream, they might be misinterpreted at the receiving end. URL encoding converts characters that are not legal in a URL to character-entity equivalents; URL decoding reverses the encoding. For example, when embedded in a block of text to

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

be transmitted in a URL, the characters < and > are encoded as %3c and %3d. The string to convert. HttpValidationStatus enumeration (System.Web) UrlEncodeUnicodeToBytes Description Provides enumerated values that indicate cache validation status. **UrlEncodeUnicodeToBytes** IgnoreThisRequest; HttpValidationStatus [C#] public const IgnoreThisRequest; HttpValidationStatus public: const [C++]HttpValidationStatus [VB] Public IgnoreThisRequest As Const HttpValidationStatus; IgnoreThisRequest [JScript] public var Description Specifies not to validate the entity in the cache. **UrlEncodeUnicodeToBytes** HttpValidationStatus Invalid; public const [C#] **HttpValidationStatus** Invalid; public: const [C++]HttpValidationStatus Invalid Public As [VB] Const HttpValidationStatus; [JScript] public Invalid var Description

Indicates that the cache is invalid.

UrlEncodeUnicodeToBytes

[C#]	public	const	HttpValidationStatus			Valid;
[C++]	public:	const		HttpValidatio	onStatus	Valid;
[VB]	Public	Const	Valid	As	HttpValida	tionStatus
[JScript]	public	var	Valid	:	HttpValidat	ionStatus;

# Description

Indicates that the cache is valid.

HttpWorkerRequest class (System.Web)

**ToString** 

# Description

This abstract class defines the base worker methods and enumerations used by ASP.NET managed code for request processing.

**ToString** 

[C#]	public	const		int	HeaderAccep	
[C++]	public:	c	onst	int	HeaderAccept	
[VB]	Public	Const	HeaderAccept		As	Integer
[JScript]	public	var	HeaderAccept		:	int;

Description

1 **ToString** 2 3 HeaderAcceptCharset; public int const [C#] HeaderAcceptCharset; int public: const [C++]Integer HeaderAcceptCharset As Public Const [VB] 6 int; HeaderAcceptCharset [JScript] public var 7 8 Description 9 10 **ToString** 11 12 HeaderAcceptEncoding; int public const [C#] 13 HeaderAcceptEncoding; public: int const [C++]14 HeaderAcceptEncoding Integer As Public Const [VB] 15 HeaderAcceptEncoding int; public [JScript] var 16 17 Description 18 19 **ToString** 20 21 HeaderAcceptLanguage; int public const [C#] 22 HeaderAcceptLanguage; public: int const [C++]23 As Integer Public Const HeaderAcceptLanguage [VB] 24 HeaderAcceptLanguage int; [JScript] public var

1	-						
2	Description						
3							
4	ToSt	ring					
5					1 <i>i</i>		
6	[C#]	public	const	int	HeaderA	AcceptRa	inges;
7	[C++]	public:	const	int	Header	AcceptRa	inges;
8	[VB]	Public C	onst He	aderAcceptRange	s As	; Ir	nteger
9	[JScript]	public	var	HeaderAcceptRa	nges	:	int;
10							
11	Description	ı					
12				•			
13	ToS	tring					
14							
15	[C#]	public	con	st int		Heade	erAge;
16	[C++]	public:	co	nst int	į	Heade	erAge;
17	[VB]	Public	Const	HeaderAge	As	I	nteger
18	[JScript]	public	var	HeaderAg	e	:	int;
19							
20	Description	n					
21							
22	ToS	String					
23							
24	[C#]	public	cons	st int		Header	Allow;
25	[C++]	public:	COI	nst int		Header	Allow;
	••						

1	[VB]	Public	Const	HeaderAllow	As	Integer
2	[JScript]	public	var	HeaderAllow	:	int;
3						
4	Description	ı				
5						
6	ToS	tring				
7						
8	[C#]	public	cons	t int	HeaderAu	thorization;
9	[C++]	public:	con	st int	HeaderAu	thorization;
10	[VB]	Public	Const	HeaderAuthorization	As	Integer
11	[JScript]	public	var	HeaderAuthorizat	ion	: int;
12						
13	Descriptio.	n				
14						
15	ToS	String				
16						
17	[C#]	public	cons	st int	HeaderCa	acheControl;
18	[C++]	public:	con	nst int	HeaderCa	acheControl;
19	[VB]	Public	Const	HeaderCacheControl	As	Integer
20	[JScript]	public	var	HeaderCacheCon	itrol	: int;
21						
22	Description	on				
23						
24	То	String				
25						

1							
2	[C#]	public	const	int	]	HeaderCo	onnection;
3	[C++]	public:	cons	st int		HeaderCo	onnection;
4	[VB]	Public	Const	HeaderConne	ection	As	Integer
5	[JScript]	public	var	HeaderCo	onnection	:	int;
6							
7	Description	n					
8							
9	ToS	String					
10							
11	[C#]	public	const	int	Head	erContent	Encoding;
12	[C++]	public:	const	int	Head	erContent	Encoding;
13	[VB]	Public	Const H	eaderContentE	ncoding	As	Integer
14	[JScript]	public	var	HeaderConte	entEncodin	g :	int;
15							
16	Description	on					
17							
18	Tos	String					
19							
20	[C#]	public	const	int	Head	erContent	Language;
21	[C++]	public:	const	int	Head	erContent	Language;
22	[VB]	Public	Const H	[eaderContentL	anguage	As	Integer
23	[JScript]	public	var	HeaderConte	entLanguag	ge :	int;
24							
25	Description	on					

1						
2	ToS	String				
3						
4	[C#]	public	const	int	Header	ContentLength;
5	[C++]	public:	cons	t int	Header	ContentLength;
6	[VB]	Public	Const	HeaderContentLe	ength A	As Integer
7	[JScript]	public	var	HeaderConte	ntLength	: int;
8						
9	Descriptio	n				
10						
11	ToS	String				
12						
13	[C#]	public	const	int	HeaderC	ontentLocation;
14	[C++]	public:	const	int	HeaderC	ontentLocation;
15	[VB]	Public	Const I	HeaderContentLo	cation	As Integer
16	[JScript]	public	var	HeaderConten	tLocation	: int;
17						
18	Description	on				
19						
20	To	String				
21						
22	[C#]	public	cons	st int		derContentMd5;
23	[C++]	public:	cor			derContentMd5;
24	[VB]	Public	Const	HeaderContent		As Integer
25	[JScript]	public	var	HeaderCon	tentMd5	: int;

1									
2	Description	ı							
3	•								
4	ToS	tring							
5									
6	[C#]	public	const	int	Header	Conten	tRange;		
7	[C++]	public:	cons	t int	Header	Conten	tRange;		
8	[VB]	Public	Const	HeaderContentR	ange A	S	Integer		
9	[JScript]	public	var	HeaderConte	ntRange	:	int;		
10									
11	Description								
12									
13	ToS	String							
14									
15	[C#]	public	cons	t int	Head	erConte	entType;		
16	[C++]	public:	cor	int int	Head	erConte	entType;		
17	[VB]	Public	Const	HeaderContent	Гуре А	S	Integer		
18	[JScript]	public	var	HeaderCont	entType	:	int;		
19									
20	Description	on							
21									
22	Tos	String							
23									
24	[C#]	public	c	onst in	nt		rCookie;		
25	[C++]	public	•	const i	nt	Heade	rCookie;		

1	[VB]	Public	Const	HeaderCookie	As	Integer
2	[JScript]	public	var	HeaderCookie	e	: int;
3						
4	Description					
5						
6	ToSt	ring				
7						
8	[C#]	public	co	nst int		HeaderDate;
9	[C++]	public:	C	onst int		HeaderDate;
10	[VB]	Public	Const	HeaderDate	As	Integer
11	[JScript]	public	var	HeaderDate		: int;
12						
13	Description	ı				
14						
15	ToSi	tring				
16	;					
17	[C#]	public	co	onst int		HeaderEtag;
18	[C++]	public:	c	const int		HeaderEtag;
19	[VB]	Public	Const	HeaderEtag	As	Integer
20	[JScript]	public	var	HeaderEtag	5	: int;
2						
22	Description	ı				
2:	3					
24	ToS	tring				
2.	5					

1						
2	[C#]	public	const	int	Head	erExpect;
3	[C++]	public:	cons	t int	Head	erExpect;
4	[VB]	Public	Const	HeaderExpect	As	Integer
5	[JScript]	public	var	HeaderExpect	:	int;
6						
7	Description					
8						
9	ToSta	ring				
10						
11	[C#]	public	const	int	Head	erExpires;
12	[C++]	public:	cons	t int	Head	erExpires;
13	[VB]	Public	Const	HeaderExpires	As	Integer
14	[JScript]	public	var	HeaderExpires	:	int;
15						
16	Description					
17						
18	ToSt	ring				
19						
20	[C#]	public	cons	t int	Не	aderFrom;
21	[C++]	public:	con	ist int	He	aderFrom;
. 22	[VB]	Public	Const	HeaderFrom	As	Integer
23	[JScript]	public	var	HeaderFrom	:	int;
24						
25	Description					

1									
2	ToS	tring							
3									
4	[C#]	public	CO	onst	int	Н	eaderHost;		
5	[C++]	public:	•	const	int		eaderHost;		
6	[VB]	Public	Const	Heade	rHost	As	Integer		
7	[JScript]	public	var	Нє	eaderHost	:	int;		
8									
9	Description	n							
10									
11	ToString								
12									
13	[C#]	public	COI	nst	int	Head	lerIfMatch;		
14	[C++]	public:	co	onst	int	HeaderIfMato			
15	[VB]	Public	Const	HeaderI	fMatch	As	Integer		
16	[JScript]	public	var	Hea	HeaderIfMatch		int;		
17									
18	Description	n							
19									
20	Tos	String							
21									
22	[C#]	public	const	int	Не	aderIfMoo	difiedSince;		
23	[C++]	public:	const	int	Не	aderIfMo	difiedSince;		
24	[VB]	Public	Const I	HeaderIfMo	difiedSince	As	Integer		
25	[JScript]	public	var	HeaderIf	ModifiedSir	nce	: int;		

11								
1								
2	Description	!						
3								
4	ToSt	tring						
5								
6	[C#]	public	const	int	He	eaderIfNo	neMatch;	
7	[C++]	public:	const	int	Н	eaderIfNo	neMatch;	
8	[VB]	Public	Const	HeaderIfNon	eMatch	As	Integer	
9	[JScript]	public	var	HeaderIf	NoneMatch	:	int;	
10								
11	Description							
12								
13	ToString							
14								
15	[C#]	public	cor	ıst	int	Heade	rIfRange;	
16	[C++]	public:	co	onst	int	Heade	rIfRange;	
17	[VB]	Public	Const	HeaderIfI	Range	As	Integer	
18	[JScript]	public	var	Heade	erIfRange	:	int;	
19								
20	Description	n						
21								
22	ToS	String						
23								
24	[C#]	public	const	int	Header	IfUnmodi	fiedSince;	
25	[C++]	public:	const	int	Header	IfUnmodi	fiedSince;	
'	•							

1	[VB]	Public (	Const	Heade	rIfUnmodifiedSince	As	Integer		
2	[JScript]	public	var	Не	aderIfUnmodifiedSi	nce :	int;		
3									
4	Descriptio	n							
5									
6	Tos	String							
7									
8	[C#]	public	c	onst	int	HeaderK	eepAlive;		
9	[C++]	public:		const	int	HeaderK	eepAlive;		
10	[VB]	Public	Const	F	IeaderKeepAlive	As	Integer		
11	[JScript]	public	va	r	HeaderKeepAlive	e :	int;		
12									
13	Description								
14									
15	То	String							
16									
17	[C#]	public	co	onst	int	HeaderLast			
18	[C++]	public:	C	const	int	HeaderLast			
19	[VB]	Public	Const	Не	eaderLastModified	As	Integer		
20	[JScript]	public	var		HeaderLastModific	ed :	int;		
21									
22	Descripti	on							
23									
24	To	String							
25									

11							
1							
2	[C#]	public	const	int		Location;	
3	[C++]	public:	const	int	Headerl	Location;	
4	[VB]	Public	Const	Const HeaderLocation		Integer	
5	[JScript]	public	var	var HeaderLocatio		int;	
6							
7	Description						
8							
9	ToString						
10							
11	[C#]	public	const	int	HeaderMaxl	Forwards;	
12	[C++]	public:	const	const int		HeaderMaxForwards;	
13	[VB]	Public	Const He	eaderMaxForwards	As	Integer	
14	[JScript]	public	var	HeaderMaxForwa	ards :	int;	
15		•					
16	Description						
17	Description						
18	ToString						
19	Tobumg						
	[C#] public		cons	const int		HeaderPragma;	
20	[C++]	public:	con			erPragma;	
21	[VB]	Public	Const	HeaderPragma	As	Integer	
22				HeaderPragm		int;	
23	[JScript]	public	var	Headell tagill	•	11109	
24							
25	Description	on .					

1							
2	ToString						
3							
4	[C#]	public	const	int	HeaderF	'roxyAutl	henticate;
5	[C++]	public:	const	int	HeaderF	'roxyAutl	henticate;
6	[VB]	Public C	Const H	IeaderProxyA	uthenticate	As	Integer
7	[JScript]	public	var	HeaderPro	xyAuthenticate	:	int;
8							
9	Description						
10							
11	ToString						
12							
13	[C#]	public	const	int	HeaderP	oxyAuth:	orization;
14	[C++]	public:	const	int	HeaderP	roxyAuth	orization;
15	[VB]	Public C	Const H	eaderProxyA	uthorization	As	Integer
16	[JScript]	public	var	HeaderPro	xyAuthorizatio	n :	int;
17							
18	Description						
19	,						
20	ToString						
21							
22	[C#]	public	1	const	int		derRange;
23	[C++]	public		const	int		derRange;
24	[VB]	Public	Const		rRange	As	Integer
25	[JScript]	public	va va	ır He	aderRange	:	int;

1								
2	Description							
3	Description							
4	ToString							
5	Tooting							
6	[C#]	public	const	int	Heade	erReferer;		
7	[C++]	public:	const	int	HeaderReferer;			
8	[VB]	Public	Const	HeaderReferer	As	Integer		
9	[JScript]	public	var	HeaderReferer	:	int;		
10								
11	Description							
12								
13	ToString							
14								
15	[C#]	public	const	int	HeaderR	etryAfter;		
16	[C++]	public:	const	int	HeaderRetryAfter			
17	[VB]	Public	Const H	eaderRetryAfter	As	Integer		
18	[JScript]	public	var	HeaderRetryAfter	:	int;		
19								
20	Description							
21								
22	ToString							
23								
24	[C#]	public	const	int		derServer;		
25	[C++]	public:	cons	t int	Hea	derServer;		

1	[VB]	Public	Const	HeaderServer	As	Integer
2	[JScript]	public	var	HeaderServer	:	int;
3						
4	Description					
5						
6	ToSt	ring				
7						
8	[C#]	public	const	int	Header	SetCookie;
9	[C++]	public:	const	int	Header	SetCookie;
10	[VB]	Public	Const	HeaderSetCookie	As	Integer
11	[JScript]	public	var	HeaderSetCookie	:	int;
12						
13	Description	1				
14						
15	ToSt	tring				
16						
17	[C#]	public	co	onst int		HeaderTe;
18	[C++]	public	: с	eonst int		HeaderTe;
19	[VB]	Public	Const	HeaderTe	As	Integer
20	[JScript]	public	var	HeaderTe	:	int;
21						
22	Description	i				
23						

1							
2	[C#]	public	COI	nst	int	Head	erTrailer;
3	[C++]	public:	CO	onst	int	Head	erTrailer;
4	[VB]	Public	Const	HeaderT	railer	As	Integer
5	[JScript]	public	var	Head	erTrailer	:	int;
6							
7	Description	$\iota$					
8							
9	ToSt	tring					
10							
11	[C#]	public	const	int	Heade	erTransferI	Encoding;
12	[C++]	public:	const	int	Heade	erTransferI	Encoding;
13	[VB]	Public Co	onst He	aderTransfer	Encoding	As	Integer
14	[JScript]	public	var	HeaderTran	sferEncodin	ig :	int;
15							
16	Description	ı					
17							
18	ToSi	tring					
19							
20	[C#]	public	con	st	int	Header	Upgrade;
21	[C++]	public:	CO	nst	int	Header	Upgrade;
22	[VB]	Public	Const	HeaderUp	grade	As	Integer
23	[JScript]	public	var	Heade	rUpgrade	:	int;
24							
25	Description	ı					

1						
2	ToSt	ring				
3						
4	[C#]	public	const	int	HeaderU	serAgent;
5	[C++]	public:	cons	t int	HeaderU	serAgent;
6	[VB]	Public	Const	HeaderUserAgent	As	Integer
7	[JScript]	public	var	HeaderUserAgent	:	int;
8						
9	Description	!				
10						
11	ToSt	ring				
12						
13	[C#]	public	cor	nst int	He	aderVary;
14	[C++]	public:	co	onst int	He	aderVary;
15	[VB]	Public	Const	HeaderVary	As	Integer
16	[JScript]	public	var	HeaderVary	:	int;
17						
18	Description	!				
19						
20	ToSt	ring				
21						
22	[C#]	public	со	nst int	Н	eaderVia;
23	[C++]	public:	c	onst int	Н	eaderVia;
24	[VB]	Public	Const	HeaderVia	As	Integer
25	[JScript]	public	var	HeaderVia	:	int;

```
Description
2
3
          ToString
5
                                                                   HeaderWarning;
    [C#]
                   public
                                     const
                                                     int
6
                                                                   HeaderWarning;
                    public:
    [C++]
                                      const
                                                     int
                                           HeaderWarning
                                                                            Integer
                Public
                              Const
                                                                  As
    [VB]
                    public
                                              HeaderWarning
    [JScript]
                                                                               int;
                                   var
10
    Description
11
12
          ToString
13
14
                                                         HeaderWwwAuthenticate;
                 public
                                              int
    [C#]
                                const
15
                  public:
                                                         HeaderWwwAuthenticate;
    [C++]
                                 const
                                              int
    [VB]
              Public
                          Const
                                     HeaderWwwAuthenticate
                                                                    As
                                                                            Integer
17
    [JScript]
                  public
                                         HeaderWwwAuthenticate
                               var
                                                                               int;
19
    Description
20
21
          ToString
22
23
                  public
                                                               ReasonCachePolicy;
    [C#]
                                   const
                                                  int
                   public:
                                                               ReasonCachePolicy;
                                    const
                                                   int
```

```
[VB]
               Public
                            Const
                                        ReasonCachePolicy
                                                                           Integer
                                                                 As
    [JScript]
                                           ReasonCachePolicy
                   public
                                                                              int;
                                 var
3
    Description
5
          ToString
    [C#]
                  public
                                                            ReasonCacheSecurity;
                                                int
                                 const
                  public:
    [C++]
                                                int
                                                            ReasonCacheSecurity;
                                  const
    [VB]
               Public
                           Const
                                       ReasonCacheSecurity
                                                                  As
                                                                           Integer
10
    [JScript]
                   public
                                          ReasonCacheSecurity
                                                                              int;
                                var
12
    Description
13
14
          ToString
15
16
    [C#]
                 public
                                              int
                                                          ReasonClientDisconnect;
                                const
                  public:
                                              int
                                                          ReasonClientDisconnect;
    [C++]
                                 const
18
              Public
                          Const
                                     ReasonClientDisconnect
    [VB]
                                                                  As
                                                                           Integer
                  public
                                         ReasonClientDisconnect:
    [JScript]
                                                                              int;
                               var
21
    Description
22
23
          ToString
24
25
```

1								
2	[C#]	public		const	int		Reason	nDefault;
3	[C++]	public	:	const	int		Reason	nDefault;
4	[VB]	Public	Const	Re	asonDefault	As	:	Integer
5	[JScript]	public	ve ve	ır	ReasonDefa	ult	:	int;
6								
7	Description	on						
8								
9	То	String						
10								
11	[C#]	public	const	int	Reas	onFileHa	ndleCa	cheMiss;
12	[C++]	public:	const	in	Reas	onFileHa	ndleCa	cheMiss;
13	[VB]	Public C	onst Re	easonFile	HandleCache	Miss	As	Integer
14	[JScript]	public	var	Reason	FileHandleCa	cheMiss	:	int;
15								
16	Description	on						
17								
18	То	String						
19								
20	[C#]	public	const	in	t Rea	sonRespo	onseCa	cheMiss;
21	[C++]	public:	const	i	nt Rea	sonRespo	onseCa	cheMiss;
22	[VB]	Public C	Const R	ReasonRe	sponseCachel	Miss	As	Integer
23	[JScript]	public	var	Reason	nResponseCa	cheMiss	:	int;
24								
25	Description	on						

**ToString** 2 3 RequestHeaderMaximum; [C#] public int const public: int RequestHeaderMaximum; [C++]const Public As Integer RequestHeaderMaximum [VB] Const RequestHeaderMaximum public int; [JScript] var 8 Description 10 **ToString** 11 12 ResponseHeaderMaximum; [C#] public const int 13 public: int ResponseHeaderMaximum; [C++]const Public ResponseHeaderMaximum As Integer [VB] Const 15 [JScript] public ResponseHeaderMaximum int; var 17 Description 18 19 HttpWorkerRequest 20 Example Syntax: 21 ToString 22 23 HttpWorkerRequest(); protected [C#] HttpWorkerRequest(); protected:

public CloseConnection(); [C#] virtual void 239 MS1-863US.APP lee@hayes plic 509+324+9256

Sub

MachineConfigPath

get MachineConfigPath();

get MachineInstallDirectory();

New()

{get;}

String:

{get;}

String;

 3.00	
State of the state	
the state of the s	

	1	[C++]	public:	virtual	void	CloseConnection();							
	2	[VB]	Overridable	Public	Sub	CloseConnection()							
	3	[JScript]	Script] public		inction	CloseConnection();							
	4												
	5	Description	ı										
	6												
	7	End	OfRequest										
	8												
	9	[C#]	public	abstract	void	EndOfRequest();							
34	10	[C++]	public: vi	rtual voi	d EndOfRe	equest() = 0;							
31 31 31 31	11	[VB]	MustOverride	Publ	ic Sub	EndOfRequest()							
Destruction of the second of t	12	[JScript]	public	abstract	function	EndOfRequest();							
4	13		ŕ										
	14	Descriptio	n										
	15	1											
d. 2.	16	    Flu	shResponse										
	17		1										
	18	    [C#]	public abstra	act void	FlushRespon	se(bool finalFlush);							
	19	FG + 13	public: virtual	void Flus	shResponse(bool	finalFlush) = $0$ ;							
	20	EXTD3 Mar	•	c Sub FlushR	esponse(ByVal	finalFlush As Boolean)							
	21	[JScript]	public abstrac		FlushResponse(f								
	22		•										
	23	D	on										
	24												
	25	C	etAppPath										

.,							
1							
2	[C#]	public	vi	rtual	string	Ge	tAppPath();
3	[C++]	public:	vi	rtual	String*	Ge	tAppPath();
4	[VB]	Overridable	Public	Function	GetApp	Path() A	s String
5	[JScript]	public	functi	ion (	GetAppPath(	:	String;
6							
7	Descript	ion					
8	W	hen overridden	in a derive	d class, ret	urns the virt	ual path to t	he currently
9	executin	g		server			application.
10	Return V	<i>alue:</i> The virtua	al path of th	e current ap	oplication.		
11	G	etAppPathTrans	slated				
12							
13	[C#]	public	virtual	stri	ng C	GetAppPath7	Translated();
14	[C++]	public:	virtual	Stri	ng* C	GetAppPath7	Translated();
15	[VB]	Overridable P	ublic Fun	ction Ge	tAppPathTra	anslated()	As String
16	[JScript]	public	function	GetAp	pPathTransla	ated() :	String;
17							
18	Descript	ion					
19	W	hen overridden	in a derive	ed class, ret	urns the UN	C-translated	l path to the
20	currently	7	executing		server		application.
21	Return V	alue: The UNC	physical pa	ath of the cu	ırrent applic	ation.	
22	G	etAppPoolID					
23							
24	[C#]	public	virt	ual	string	GetA	ppPoolID();
25	[C++]	public:	vir	ual	String*	GetA	ppPoolID();

```
[VB]
             Overridable
                             Public
                                        Function
                                                    GetAppPoolID()
                                                                                String
                                                                         As
    [JScript]
                   public
                                 function
                                                GetAppPoolID()
                                                                                String;
2
3
    Description
           When overridden in a derived class, returns the application pool ID for the
5
                                                                                 URL.
    current
6
    Return Value: Always returns null.
7
           GetBytesRead
8
9
    [C#]
                   public
                                     virtual
                                                      long
                                                                      GetBytesRead();
10
    [C++]
                    public:
                                    virtual
                                                      int64
                                                                      GetBytesRead();
11
              Overridable
    [VB]
                              Public
                                        Function
                                                      GetBytesRead()
                                                                          As
                                                                                 Long
12
                                  function
                                                  GetBytesRead()
    [JScript]
                    public
                                                                                 long;
13
14
    Description
15
16
           GetClientCertificate
17
18
                  public
                                 virtual
                                                                GetClientCertificate();
    [C#]
                                                 byte[]
19
    [C++]
             public:
                        virtual
                                  unsigned
                                              char
                                                      GetClientCertificate()
                                                                               gc[];
20
                           Public
    [VB]
            Overridable
                                     Function
                                                 GetClientCertificate()
                                                                          As
                                                                                Byte()
21
                                             GetClientCertificate()
    [JScript]
                               function
                  public
                                                                               Byte[];
22
23
    Description
24
25
```

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

]	Defines the base worker class used by ASP.NET managed code for request
process	sing.
(	GetClientCertificateRinaryIssuer

[C#] public virtual byte[] GetClientCertificateBinaryIssuer();
[C++] public: virtual unsigned char GetClientCertificateBinaryIssuer() \_\_gc[];
[VB] Overridable Public Function GetClientCertificateBinaryIssuer() As Byte()
[JScript] public function GetClientCertificateBinaryIssuer() : Byte[];

Description

### GetClientCertificateEncoding

GetClientCertificateEncoding(); public virtual int [C#] virtual GetClientCertificateEncoding(); public: int [C++][VB] Overridable Public Function GetClientCertificateEncoding() As Integer [JScript] public function GetClientCertificateEncoding() int;

Description

### GetClientCertificatePublicKey

[C#] public virtual byte[] GetClientCertificatePublicKey();
[C++] public: virtual unsigned char GetClientCertificatePublicKey() \_\_gc[];
[VB] Overridable Public Function GetClientCertificatePublicKey() As Byte()

[JScript] GetClientCertificatePublicKey() public function Byte[]; 2 Description 3 **GetClientCertificateValidFrom** 5 6 [C#] public virtual DateTime GetClientCertificateValidFrom(); DateTime GetClientCertificateValidFrom(); [C++]public: virtual 8 [VB] Overridable Public Function GetClientCertificateValidFrom() As DateTime [JScript] public function GetClientCertificateValidFrom() : DateTime: 10 11 Description 12 13 **GetClientCertificateValidUntil** 14 15 [C#] public virtual DateTime GetClientCertificateValidUntil(); 16 [C++]DateTime GetClientCertificateValidUntil(); public: virtual 17 [VB] Overridable Public Function GetClientCertificateValidUntil() As DateTime 18 GetClientCertificateValidUntil() [JScript] public function DateTime; 19 20 Description 21 22 GetConnectionID 23 24 [C#] public virtual long GetConnectionID();

1 [C++]public: virtual int64 GetConnectionID(); [VB] Overridable Public Function GetConnectionID() As Long 2 [JScript] public function GetConnectionID() long; 3 4 Description 5 When overridden in a derived class, returns the ID of the current 6 connection. 7 Return Value: Always returns 0. 8 GetFilePath 9 10 public [C#] virtual string GetFilePath(); 11 [C++]public: virtual String\* GetFilePath(); 12 [VB] Overridable Public Function GetFilePath() As String 13 [JScript] public function GetFilePath() String; 14 15 Description 16 When overridden in a derived class, returns the physical path to the 17 requested URI. 18 Return Value: The path to the URI. 19 GetFilePathTranslated 20 21 [C#] public virtual string GetFilePathTranslated(): 22 String\* [C++]public: virtual GetFilePathTranslated(); 23 Overridable Public Function [VB] GetFilePathTranslated() As String 24 [JScript] public function GetFilePathTranslated() String; 25

1

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

When overridden in a derived class, returns the translated file path to the requested URI (from virtual path to UNC path, ie "/proj1/page.aspx" to "c:\dir\page.aspx")

Return Value: The translated file path.

#### GetHttpVerbName

GetHttpVerbName(); public abstract string [C#] public: String\* GetHttpVerbName() virtual 0; [C++]MustOverride Public Function GetHttpVerbName() [VB] As String [JScript] public abstract function GetHttpVerbName() String;

### Description

Provides access to the specified member of the request header.

Return Value: The HTTP verb returned in the request header.

#### **GetHttpVersion**

public GetHttpVersion(); [C#] abstract string public: [C++]virtual String\* GetHttpVersion() 0; [VB] MustOverride Public Function GetHttpVersion() String As [JScript] public abstract function GetHttpVersion() String;

### Description

lee@hayes pilc 509+324+9256 246 MS1-863US.APP

Provides access to the specified member of the request header. 1 Return Value: The HTTP version returned in the request header. GetKnownRequestHeader 3 4 public virtual string GetKnownRequestHeader(int index); [C#] String\* GetKnownRequestHeader(int [C++]public: virtual index); [VB] Overridable Public Function GetKnownRequestHeader(ByVal index As Integer) As 8 public function GetKnownRequestHeader(index : int) : String: [JScript] 10 Description 11 12 GetKnownRequestHeaderIndex 13 14 GetKnownRequestHeaderIndex(string header); [C#] public static int 15 GetKnownRequestHeaderIndex(String\* header); [C++]public: static int 16 [VB] Public Shared Function GetKnownRequestHeaderIndex(ByVal header As 17 String) As Integer 18 [JScript] public static function GetKnownRequestHeaderIndex(header: String): 19 int; 20 21 Description 22 23 GetKnownRequestHeaderName 24

String

[C#]	public	static	string	GetKnownRequestHeaderName(int	index);
[C++]	public:	static	String*	GetKnownRequestHeaderName(int	index);
[VB] F	Public Sha	ared Fu	nction Ge	tKnownRequestHeaderName(ByVal in	ndex As
Integer	)			As	String
[JScrip	t] public	static	function (	GetKnownRequestHeaderName(index	: int) :
String;					

# GetKnownResponseHeaderIndex

[C#]	public	static	int	GetKnownResponseHeaderIndex(string	header);
[C++]	public:	static	int	GetKnownResponseHeaderIndex(String*	header);
[VB] I	Public Sha	ared Fur	nction	GetKnownResponseHeaderIndex(ByVal h	eader As
String)				As	Integer
[JScrip	t] public	static fu	nctio	n GetKnownResponseHeaderIndex(header:	String):
int;					

## Description

# GetKnownResponseHeaderName

[C#] public static string GetKnownResponseHeaderName(int index);
[C++] public: static String\* GetKnownResponseHeaderName(int index);

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

```
[VB] Public Shared Function GetKnownResponseHeaderName(ByVal index As
                                                                         String
                                       As
Integer)
[JScript] public static function GetKnownResponseHeaderName(index : int) :
String;
Description
      GetLocalAddress
                                                            GetLocalAddress();
                             abstract
                                             string
             public
[C#]
                                  String*
                                               GetLocalAddress()
                                                                             0;
           public:
                       virtual
[C++]
                                  Function
                                              GetLocalAddress()
                                                                   As
                                                                         String
        MustOverride
                        Public
[VB]
            public
                                  function
                                             GetLocalAddress()
                                                                        String;
[JScript]
                      abstract
Description
      Provides access to the specified member of the request header.
Return Value: The server's IP address returned in the request header.
       GetLocalPort
                                                  int
                                                                GetLocalPort();
               public
                                abstract
[C#]
                                                GetLocalPort()
                                                                             0;
             public:
                          virtual
                                       int
[C++]
                                                GetLocalPort()
                                                                        Integer
         MustOverride
                          Public
                                    Function
                                                                  As
[VB]
[JScript]
              public
                         abstract
                                     function
                                                   GetLocalPort()
                                                                            int;
Description
```

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Provides access to the specified member of the request header.

Return Value: The server's port number returned in the request header.

#### GetPathInfo

[C#]	public	vi	rtual	string	GetPa	thInfo();
[C++]	public:	V	irtual	String*	GetPa	thInfo();
[VB]	Overridable	Public	Function	GetPathInfo()	As	String
[JScript]	public	funct	ion G	etPathInfo()	:	String;

#### Description

When overridden in a derived class, returns additional path information for a resource with a URL extension. That is, for the URL /virdir/page.html/tail, the PathInfo value is /tail.

Return Value: Additional path information for a resource.

### GetPreloadedEntityBody

GetPreloadedEntityBody(); virtual byte[] [C#] public unsigned char GetPreloadedEntityBody() gc[]; public: virtual [C++]Overridable Public Function GetPreloadedEntityBody() As Byte() [VB] GetPreloadedEntityBody() Byte[]; public function [JScript]

#### Description

#### **GetProtocol**

i,	34. 34.
١,	3
ij	20 20 20
:	# #
ľ	
í,	2
i,	⊋: 3:
	3) 3)
íŝ	
ï,	3: 2:
١;	4
ŀ	<u>}</u> :
ĺ,	3
į,	******
ŀ	

1		
2	[C#] public virtual string GetProtoc	col();
3	[C++] public: virtual String* GetProtoc	col();
4	[VB] Overridable Public Function GetProtocol() As S	String
5	[JScript] public function GetProtocol() : S	tring;
6		
7	Description	
8	When overridden in a derived class, returns the HTTP protocol (HT	TP or
9	HTTPS).	
10	Return Value: HTTPS if IsSecure is true, otherwise false.	
11	GetQueryString	
12		
13	[C#] public abstract string GetQueryStr	ring();
14	[C++] public: virtual String* GetQueryString() =	0;
15	[VB] MustOverride Public Function GetQueryString() As	String
16	[JScript] public abstract function GetQueryString() : S	String;
17		
18	Description	
19	Provides access to the specified member of the request he	eader.
20	Return Value: The QueryString member of the request header.	
21	GetQueryStringRawBytes	
22		
23	[C#] public virtual byte[] GetQueryStringRawBy	ytes();
24	[C++] public: virtual unsigned char GetQueryStringRawBytes() _	_gc[];
25	[VB] Overridable Public Function GetQueryStringRawBytes() As	Byte()

	1	[JScript]	public	function	GetQue	ryStringRawBytes()	:	Byte[];			
	2										
	3	Description									
	4	When overridden in a derived class, returns the response query string as an									
	5	array of bytes.									
	6	Return Value: An array of bytes containing the response.									
	7	Getl	RawUrl								
	8										
	9	[C#]	public	а	bstract	string	GetRa	awUrl();			
	10	[C++]	public:	virtual	String	* GetRawUrl()	) =	0;			
	11	[VB] N	//ustOverride	e Public	Functi	on GetRawUrl()	As	String			
100 min	12	[JScript]	public	abstract	functio	n GetRawUrl()	:	String;			
	13										
	14	Description									
	15	Provides access to the specified member of the request header.									
	16	Return Value: The raw url member of the request header.									
:	17	GetRemoteAddress									
	18										
	19	[C#]	public	abst	ract	string GetF	RemoteA	ddress();			
	20	[C++]	public:	virtual	String*	GetRemoteAddre	ss()	= 0;			
	21	[VB] M	ustOverride	Public	Function	GetRemoteAddress	s() As	String			
	22	[JScript]	public	abstract	function	GetRemoteAddress	s() :	String;			
	23										
	24	Description	on								
	25										

Provides access to the specified member of the request header. 1 Return Value: The client's IP address returned in the request header. GetRemoteName 3 public virtual string GetRemoteName(); [C#] [C++]public: String\* GetRemoteName(); virtual Overridable [VB] Public Function GetRemoteName() As String [JScript] public function GetRemoteName() String; 9 Description 10 When overridden in a derived class, returns the client computer's name. 11 Return Value: The name of the client machine. 12 GetRemotePort 13 14 [C#] public abstract GetRemotePort(); int 15 [C++]public: virtual int GetRemotePort() 0; MustOverride [VB] Public Function GetRemotePort() As Integer 17 [JScript] public abstract function GetRemotePort() int; 18 19 Description 20 Provides access to the specified member of the request header. 21 Return Value: The client HTTP port number returned in the request header. 22 GetRequestReason 23 24 public [C#] virtual int GetRequestReason();

lee **②**hayes ølc 509-324-9256 253 *MS1-863US.APP* 

20

21

22

23

24

25

[C++]GetRequestReason(); public: virtual int Overridable Public Function GetRequestReason() [VB] As Integer function [JScript] public GetRequestReason() int; Description When overridden in a derived class, returns the reason for the request. 6 Return Value: Reason code. The default is ReasonResponseCacheMiss. GetServerName 8 9 [C#] public virtual string GetServerName(); 10 String\* GetServerName(); [C++]public: virtual 11 [VB] Overridable Public **Function** GetServerName() As String 12 [JScript] public function GetServerName() String; 13 14 Description 15 When overridden in a derived class, returns the name of the local server. 16 Return Value: The name of the server. 17 **GetServerVariable** 18

[C#] public virtual string GetServerVariable(string name); [C++]public: String\* GetServerVariable(String\* virtual name); [VB] Overridable Public Function GetServerVariable(ByVal name As String) As String [JScript] public function GetServerVariable(name : String) String;

lee@hayes pik 509-324-9256 254 MS1-863US.APP

1

3

4

5

6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

### GetStatusDescription

[C#] public static string GetStatusDescription(int code);
[C++] public: static String\* GetStatusDescription(int code);
[VB] Public Shared Function GetStatusDescription(ByVal code As Integer) As
String

[JScript] public static function GetStatusDescription(code : int) : String;

### Description

# GetUnknownRequestHeader

Get Unknown Request Header (stringname); string [C#] public virtual GetUnknownRequestHeader(String\* name); String\* public: virtual [VB] Overridable Public Function GetUnknownRequestHeader(ByVal name As String As String) [JScript] public function GetUnknownRequestHeader(name : String) : String;

# Description

# GetUnknownRequestHeaders

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

```
[C#]
          public
                      virtual
                                                 GetUnknownRequestHeaders();
                                  string[][]
                               String*
                     virtual
                                           GetUnknownRequestHeaders()
[C++]
          public:
                                                                             [];
[VB] Overridable Public Function GetUnknownRequestHeaders() As String()()
[JScript]
           public
                     function
                                GetUnknownRequestHeaders()
                                                                      String[][];
Description
      GetUriPath
               public
                                                  string
                                                                  GetUriPath();
[C#]
                                abstract
                                      String*
            public:
                         virtual
                                                   GetUriPath()
                                                                             0;
[C++]
[VB]
         MustOverride
                           Public
                                     Function
                                                  GetUriPath()
                                                                          String
                                                                   As
[JScript]
             public
                        abstract
                                     function
                                                  GetUriPath()
                                                                         String;
Description
                   the
                                                     the
                                                                           URI.
      Returns
                          physical
                                      path
                                               to
                                                             requested
Return Value: The path of the requested URI.
      GetUrlContextID
[C#]
              public
                              virtual
                                              long
                                                            GetUrlContextID();
[C++]
              public:
                              virtual
                                             int64
                                                            GetUrlContextID();
[VB]
         Overridable
                        Public
                                  Function
                                              GetUrlContextID()
                                                                    As
                                                                          Long
[JScript]
               public
                            function
                                          GetUrlContextID()
                                                                          long;
```

When overridden in a derived class, returns the context ID of the current connection.

Return Value: Always returns 0.

GetUserToken

[C#]	public	vi	rtual	IntPtr	GetUse	rToken();
[C++]	public:	V	rirtual	IntPtr	GetUse	rToken();
[VB]	Overridable	Public	Function	on GetUserT	oken() As	IntPtr
[JScript]	public	func	tion	GetUserToken	:	IntPtr;

## Description

When overridden in a derived class, returns the client's impersonation token.

Return Value: The value representing the impersonation token. The default is 0.

### GetVirtualPathToken

[C#]	public	vir	tual	IntPtr	GetVirtua	lPathT	oken();
[C++]	public:	vi	irtual	IntPtr	GetVirtua	lPathT	oken();
[VB]	Overridable	Public	Function	n GetVirtu	alPathToken()	As	IntPtr
[JScript	t] public	funct	tion	GetVirtualPa	thToken()	:	IntPtr;

# Description

1	На	sEntityBody						
2								
3	[C#]	pub	olic	boo	ol	На	sEntit	yBody();
4	[C++]	pu	blic:	bo	ol	На	sEntit	yBody();
5	[VB]	Public	Function	HasEr	ntityBody()	As		Boolean
6	[JScript]	public	function	Has	EntityBody()	:		Boolean;
7								
8	Descripti	Con						
9								
10	H	eadersSent						
11								<b>~</b>
12	[C#]	public	virt		bool			ersSent();
13	[C++]	public:		rtual	bool			ersSent();
14	[VB]	Overridable		unction	HeadersSen	t()	As	Boolean
15	[JScript]	public	function	ı H	eadersSent()	:		Boolean;
16								
17	Descript	ion						
18		-						
19	Is Is	sClientConnecte	ed					
20	 	1 1'		1	bool	IcCli	entCo	nnected();
21	[C#]	public	virtua		bool			nnected();
22		public:	virtu Daldia Esse				As	Boolean
23					sClientConnected(			Boolean;
24	[JScript]	] public	function	ISCII	eniconnecieu(	J	•	Dooroan,
25	5							

2

3

5

6

9

10

11

12

13

14

15

16

17

18

19

20

21

22

24

25

IsEntireEntityBodyIsPreloaded

[C#] public virtual bool IsEntireEntityBodyIsPreloaded();
[C++] public: virtual bool IsEntireEntityBodyIsPreloaded();
[VB] Overridable Public Function IsEntireEntityBodyIsPreloaded() As Boolean
[JScript] public function IsEntireEntityBodyIsPreloaded() : Boolean;

Description

**IsSecure** 

public virtual [C#] bool IsSecure(); public: virtual bool IsSecure(); [C++]Overridable [VB] Public Function IsSecure() Boolean As [JScript] public function IsSecure() Boolean;

Description

When overridden in a derived class, returns a value indicating whether the connection is secure (using SSL).

Return Value: true if the connection is secure, otherwise false. The default is false.

MapPath

virtualPath); MapPath(string virtual string [C#] public MapPath(String\* virtualPath); String\* [C++]public: virtual [VB] Overridable Public Function MapPath(ByVal virtualPath As String) As String String; String) MapPath(virtualPath function [JScript] public

Description

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

### ReadEntityBody

size); buffer, int ReadEntityBody(byte[] int virtual public [C#] [C++] public: virtual int ReadEntityBody(unsigned char buffer \_\_gc[], int size); [VB] Overridable Public Function ReadEntityBody(ByVal buffer() As Byte, Integer Integer) As As size ByVal [JScript] public function ReadEntityBody(buffer : Byte[], size : int) : int;

Description

## Send Calculated Content Length

[C#] public virtual void SendCalculatedContentLength(int contentLength);
[C++] public: virtual void SendCalculatedContentLength(int contentLength);
[VB] Overridable Public Sub SendCalculatedContentLength(ByVal contentLength
As

1	[JScript] public function SendCalculatedContentLength(contentLength : int);							
2								
3	Description							
4								
5	SendKnownResponseHeader							
6								
7	[C#] public abstract void SendKnownResponseHeader(int index, string value);							
8	[C++] public: virtual void SendKnownResponseHeader(int index, String* value) =							
9	0;							
10	[VB] MustOverride Public Sub SendKnownResponseHeader(ByVal index As							
11	Integer, ByVal value As String)							
12	[JScript] public abstract function SendKnownResponseHeader(index : int, value :							
13	String);							
14								
15	Description							
16								
17	SendResponseFromFile							
18								
19	[C#] public abstract void SendResponseFromFile(IntPtr handle, long offset, long							
20	length);							
21	[C++] public: virtual void SendResponseFromFile(IntPtr handle,int64 offset,							
22	int64 length) = 0;							
23	[VB] MustOverride Public Sub SendResponseFromFile(ByVal handle As IntPtr,							
24	ByVal offset As Long, ByVal length As Long)							
25	[JScript] public abstract function SendResponseFromFile(handle : IntPtr, offset :							

length long); long, 2 Description 3 SendResponseFromFile 6 [C#] public abstract void SendResponseFromFile(string filename, long offset, long length); [C++] public: virtual void SendResponseFromFile(String\* filename, int64 offset, int64 length) 0; 10 [VB] MustOverride Public Sub SendResponseFromFile(ByVal filename As 11 String, ByVal offset As ByVal length As Long, Long) 12 [JScript] public abstract function SendResponseFromFile(filename: String, offset 13 length long, long); 14 15 Description 16 17 SendResponseFromMemory 18 19 [C#] public abstract void SendResponseFromMemory(byte[] data, int length); 20 [C++] public: virtual void SendResponseFromMemory(unsigned char data gc[], length) int 0; 22 [VB] MustOverride Public Sub SendResponseFromMemory(ByVal data() As ByVal Byte, length As Integer)

[JScript] public abstract function SendResponseFromMemory(data : Byte[], length

1	int);
2	
3	Description
4	
5	SendResponseFromMemory
6	
7	[C#] public virtual void SendResponseFromMemory(IntPtr data, int length);
8	[C++] public: virtual void SendResponseFromMemory(IntPtr data, int length);
9	[VB] Overridable Public Sub SendResponseFromMemory(ByVal data As IntPtr,
10	ByVal length As Integer)
11	[JScript] public function SendResponseFromMemory(data : IntPtr, length : int);
12	
13	Description
14	
15	SendStatus
16	
17	[C#] public abstract void SendStatus(int statusCode, string statusDescription);
18	[C++] public: virtual void SendStatus(int statusCode, String* statusDescription) =
19	0;
20	[VB] MustOverride Public Sub SendStatus(ByVal statusCode As Integer, ByVal
21	statusDescription As String)
22	[JScript] public abstract function SendStatus(statusCode : int, statusDescription :
23	String);
24	
25	Description

3

10

11

12

13

14

15

16

17

19

20

22

23

## SendUnknownResponseHeader

[C#] public abstract void SendUnknownResponseHeader(string name, string value);

[C++] public: virtual void SendUnknownResponseHeader(String\* name, String\* value) = 0;

[VB] MustOverride Public Sub SendUnknownResponseHeader(ByVal name As String, ByVal value As String)

[JScript] public abstract function SendUnknownResponseHeader(name : String, value : String);

Description

#### SetEndOfSendNotification

[C#] public virtual void SetEndOfSendNotification(HttpWorkerRequest.EndOfSendNotification callback, object extraData); virtual void [C++]public: SetEndOfSendNotification(HttpWorkerRequest.EndOfSendNotification\* callback, Object\* extraData); [VB] Overridable Public Sub SetEndOfSendNotification(ByVal callback As HttpWorkerRequest.EndOfSendNotification, ByVal extraData As Object) [JScript] public function SetEndOfSendNotification(callback

HttpWorkerRequest.EndOfSendNotification, Object); extraData 2 Description 3 HttpWriter class (System.Web) **ToString** 7 8 Description 9 Provides a System.IO.TextWriter object that is accessed through the 10 intrinsic System. Web. HttpResponse object. 11 The System.Web.HttpResponse.Write(System.String) methods of the 12 intrinsic System. Web. HttpResponse object make internal calls to an HttpWriter 13 object. 14 Encoding 15 **ToString** 16 17 18 Description 19 Gets an System.Text.Encoding object for the System.IO.TextWriter. 20 FormatProvider 21 NewLine 22 OutputStream 23 **ToString** 24 25

Gets a  $\mathbf{System.IO.Stream}$  object to enable HTTP output directly from the  $\mathbf{Stream}$ .

Close

[C#]	public	override	void	Close();
[C++]	public:		void	Close();
[VB]	Overrides	Public	Sub	Close()
[JScript]	public	override	function	Close();

## Description

Sends all buffered output to the HTTP output stream and closes the socket connection.

Flush

ļ	[C#]	public	override	void	Flush();
	[C++]	public:		void	Flush();
	[VB]	Overrides	Public	Sub	Flush()
	[JScript]	public	override	function	Flush();

# Description

Sends all buffered output to the HTTP output stream.

Write

[C#]	public	overrio	de	void	W	rite(cha	ır	cł	1);
[C++]	public:		void	Wri	ite(w	char_t		cł	1);
[VB]	Overrides	Public	Sub	Write(By	Val	ch	As	Cha	ır)
[JScript]	public overrid	e function	Write(ch	: Char);	Sends	HTTP	output	to t	he
client.									

2

5

6

7

8

9

10

11

12

13

15

16

17

18

19

20

21

22

23

24

25

Sends a single character to the HTTP output stream. The character to send to the HTTP output stream.

Write

```
Write(object
[C#]
                                          void
                                                                          obj);
            public
                          override
                                                  Write(Object*
[C++]
                                                                          obj);
                                  void
                 public:
[VB]
         Overrides
                                         Write(ByVal
                                                                        Object)
                       Public
                                 Sub
                                                          obi
                                                                 As
[JScript]
             public
                                      function
                                                   Write(obj
                                                                       Object);
                         override
```

### Description

Sends an **Object** to the HTTP output stream. The **Object** to send to the HTTP output stream.

Write

[C#]	public	over	ride	void	Write(	string	s);
[C++]	pub	lic:	void	Write	e(Strin	g*	s);
[VB]	Overrides	Public	Sub	Write(ByVal	S	Ας	String)

[JScript] public override function Write(s : String);

Description

Sends a string to the HTTP output stream. The string to send to the HTTP output stream.

Write

[C#] public override void Write(char[] buffer, int index, int count);
[C++] public: void Write(\_wchar\_t buffer \_\_gc[], int index, int count);
[VB] Overrides Public Sub Write(ByVal buffer() As Char, ByVal index As
Integer, ByVal count As Integer)
[JScript] public override function Write(buffer : Char[], index : int, count : int);

# Description

Sends a stream of characters with the specified starting position and number of characters to the HTTP output stream. The memory buffer containing the characters to send to the HTTP output stream The buffer position of the first character to send. The number of characters to send beginning at the position specified by *index*.

WriteBytes

[C#] public void WriteBytes(byte[] buffer, int index, int count);
[C++] public: void WriteBytes(unsigned char buffer \_\_gc[], int index, int count);
[VB] Public Sub WriteBytes(ByVal buffer() As Byte, ByVal index As Integer,
ByVal count As Integer)

[JScript] public function WriteBytes(buffer : Byte[], index : int, count : int);

# Description

2

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

25

Sends a stream of bytes with the specified starting position and number of bytes to the HTTP output stream. The memory buffer containing the bytes to send to the HTTP output stream. The buffer position of the first byte to send. The number of bytes to send, beginning at the byte position specified by *index*.

#### WriteLine

[C#]	public	override	void	WriteLine();
[C++]	public:		void	WriteLine();
[VB]	Overrides	Public	Sub	WriteLine()
[JScript]	public	override	function	WriteLine();

## Description

Sends a carriage return + line feed (CRLF) pair of characters to the HTTP output stream.

#### WriteString

[C#] public void WriteString(string int index, int count); [C++]public: void WriteString(String\* int index, int s, count); [VB] Public Sub WriteString(ByVal s As String, ByVal index As Integer, ByVal count As Integer) [JScript] public function WriteString(s: String, index: int, count: int);

lee@hayes plic 509+324+9256

Des	crip	tion
200	v, vp	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Sends a string or a sub-string to the client.

IHttpAsyncHandler interface (System.Web)

WriteString

#### Description

When implemented by a class, defines the contract that HTTP asynchronous handler objects must implement.

BeginProcessRequest

[C#] IAsyncResult BeginProcessRequest(HttpContext context, AsyncCallback cb, object extraData);

[C++] IAsyncResult\* BeginProcessRequest(HttpContext\* context,

AsyncCallback\* cb, Object\* extraData);

[VB] Function BeginProcessRequest(ByVal context As HttpContext, ByVal cb As

AsyncCallback, ByVal extraData As Object) As IAsyncResult

[JScript] function BeginProcessRequest(context : HttpContext, cb :

AsyncCallback, extraData : Object) : IAsyncResult;

#### Description

Initiates an asynchronous call to the HTTP handler.

Return Value: An System.IAsyncResult that contains information about the status of the process. An System.Web.HttpContext object that provides references to

intrinsic server objects (for example, Request, Response, Session, and Server) used to service HTTP requests. The System.AsyncCallback to call when the asynchronous method call is complete. If cb is a null reference (Nothing in Visual Basic), the delegate is not called. Any extra data needed to process the request.

# EndProcessRequest

result);	cResult	iest(IAsyn	EndProcessRequ	void		[C#]
result);	Result*	est(IAsyno	EndProcessRequ	void	-]	[C++]
IAsyncResult)	As	result	ocessRequest(ByVal	ıb EndPro	Sı	[VB]
IAsyncResult);	:	st(result	EndProcessReques	function	ript]	[JScript]

#### Description

Executes clean-up code when the process ends. An **System.IAsyncResult** that contains information about the status of the process.

IHttpHandler interface (System.Web)

EndProcessRequest

#### Description

Defines the contract that ASP.NET implements to synchronously process HTTP Web requests using custom HTTP handlers.

You can write custom HTTP handlers to process specific, predefined types of HTTP requests in any Common Language Specification (CLS) compliant language. Executable code defined in the **HttpHandler** classes, rather than conventional ASP or ASP.NET Web pages, responds to these specific requests.

HTTP handlers give you a means of interacting with the low-level request and response services of the IIS Web server and provide functionality much like ISAPI extensions but with a simpler programming model.

IsReusable

EndProcessRequest

[C#]	bo	ool		IsReusable		{get;}
[C++]		bool			get_I	sReusable();
[VB]	ReadOnly	Property		IsReusable	As	Boolean
[JScript]	abstract	function	get	IsReusable()	:	Boolean;

# Description

Gets a value indicating whether another request can use the System.Web.IHttpHandler instance.

You explicitly set the IsReusable property to true or false by code you provide that overrides the IsReusable property accessor.

ProcessRequest

[C#]		void	ProcessReq	uest(HttpCon	text	context);
[C++]		void	ProcessReq	uest(HttpCon	text*	context);
[VB]	Sub	Proc	essRequest(ByVal	context	As	HttpContext)
[JScript]		function	ProcessReque	st(context	:	HttpContext);

Description

Enables processing of HTTP Web requests by a custom **HttpHandler** that implements the **System.Web.IHttpHandler** interface.

Place your custom HttpHandler code in the ProcessRequest virtual method as shown in the following example. An System.Web.HttpContext object that provides references to the intrinsic server objects (for example, Request, Response, Session, and Server) used to service HTTP requests.

IHttpHandlerFactory interface (System.Web)

ProcessRequest

## Description

Defines the contract that class factories must implement to create new System.Web.IHttpHandler objects.

A class that implements the **IHttpHandlerFactory** interface has no behavior except to dynamically manufacture new handler objects--that is, new instances of classes that implement the **IHttpHandler** interface.

GetHandler

[C#] IHttpHandler GetHandler(HttpContext context, string requestType, string url, string pathTranslated);
[C++] IHttpHandler\* GetHandler(HttpContext\* context, String\* requestType, String\* url, String\* pathTranslated);
[VB] Function GetHandler(ByVal context As HttpContext, ByVal requestType As String, ByVal url As String, ByVal pathTranslated As String) As IHttpHandler
[JScript] function GetHandler(context: HttpContext, requestType: String, url:

String, pathTranslated : String) : IHttpHandler;

Description

2

3

4

5

6

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Return Value: A new IHttpHandler object that processes the request. An instance of the System.Web.HttpContext class that provides references to intrinsic server objects (For example, Request, Response, Session, and Server) used to service HTTP requests. The HTTP data transfer method (GET or POST) that the client uses. The System.Web.HttpRequest.RawUrl of the requested resource. The System.Web.HttpRequest.PhysicalApplicationPath to the requested resource.

ReleaseHandler

[C#] void ReleaseHandler(IHttpHandler handler);

[C++] void ReleaseHandler(IHttpHandler\* handler);

[VB] Sub ReleaseHandler(ByVal handler As IHttpHandler)

[JScript] function ReleaseHandler(handler : IHttpHandler);

Description

Enables a factory to reuse an existing handler instance. The IHttpHandler object to reuse.

IHttpModule interface (System.Web)

ReleaseHandler

Description

1	Prov	vides m	odule initial	ization	and disposal e	events to th	e inheriting class.
2	Dis	pose					
3							
4	[C#]			,	void		Dispose();
5	[C++]				void		Dispose();
6	[VB]				Sub		Dispose()
7	[JScript]			:	function		Dispose();
8							
9	Description	n					
10	Dis	poses o	of the resou	rces (o	ther than mer	nory) used	by the module that
11	implements IHttpModule.						
12	Dis	pose pe	erforms any	final c	eleanup work	prior to rea	noval of the module
13	from the ex	xecution	n pipeline.				
14	Init						
15							
16	[C#]		void		Init(HttpApp	olication	context);
17	[C++]		void		Init(HttpApp	lication*	context);
18	[VB]	Sub	Init(B)	yVal	context	As	HttpApplication)
19	[JScript]	1	function	In	it(context	:	HttpApplication);
20							
21	Description						
22	Initializes a module and prepares it to handle requests.						
23		cessInfo	o class (Syst	em.We	b)		
24	Init						
25							

13

14

15

16

17

18

19

20

21

23

24

Provides information on processes currently executing. ProcessInfo Example Syntax: Init 8 [C#] 10

Description

2

public

ProcessInfo();

public: ProcessInfo();  $\{C++\}$ 

Public Sub New() [VB]

function [JScript] public ProcessInfo();

# Description

Initializes a new instance of the System. Web. ProcessInfo class.

The default constructor initializes all fields to their default values.

ProcessInfo

Example Syntax:

Init

[C#] public ProcessInfo(DateTime startTime, TimeSpan age, int processID, int requestCount, ProcessStatus status, ProcessShutdownReason shutdownReason, int peakMemoryUsed);

[C++] public: ProcessInfo(DateTime startTime, TimeSpan age, int processID, int requestCount, ProcessStatus status, ProcessShutdownReason shutdownReason, int

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

peakMemoryUsed);

[VB] Public Sub New(ByVal startTime As DateTime, ByVal age As TimeSpan, ByVal processID As Integer, ByVal requestCount As Integer, ByVal status As ProcessStatus, ByVal shutdownReason As ProcessShutdownReason, ByVal peakMemoryUsed As Integer)

[JScript] public function ProcessInfo(startTime : DateTime, age : TimeSpan, processID : int, requestCount : int, status : ProcessStatus, shutdownReason : ProcessShutdownReason, peakMemoryUsed : int); Initializes a new instance of the System.Web.ProcessInfo class.

## Description

Initializes a new instance of the **ProcessInfo** class and sets internal information indicating the status of the process. A **System.DateTime** that indicates the time at which the process started. The **System.TimeSpan** that indicates the time elapsed since the process started. The ID number assigned to the process. The number of start requests for the process. One of the **System.Web.ProcessStatus** values that indicates the current status of the process. One of the **System.Web.ProcessShutdownReason** values. The maximum memory used (in bytes).

Age

Init

[C#] public TimeSpan Age {get;} public: TimeSpan [C++]property get Age(); [VB] Public ReadOnly Property Age As TimeSpan

```
[JScript]
                  public
                                                                         TimeSpan;
                              function
                                                      Age()
                                             get
2
    Description
          Gets the length of time the process has been running.
          PeakMemoryUsed
          Init
    [C#]
                   public
                                                 PeakMemoryUsed
                                    int
                                                                              {get;}
    [C++]
                                                           get PeakMemoryUsed();
                 public:
                                                 int
                               property
            Public
                                                PeakMemoryUsed
    [VB]
                       ReadOnly
                                    Property
                                                                      As
                                                                             Integer
10
    [JScript]
                 public
                            function
                                          get
                                                  PeakMemoryUsed()
                                                                                int;
11
12
    Description
13
          Gets the maximum amount of memory the process has used.
14
          ProcessID
15
          Init
16
17
                     public
    [C#]
                                        int
                                                       ProcessID
                                                                              {get;}
18
    [C++]
                   public:
                                                       int
                                                                   get ProcessID();
                                   __property
    [VB]
              Public
                         ReadOnly
                                        Property
                                                      ProcessID
                                                                     As
                                                                             Integer
20
    [JScript]
                   public
                               function
                                              get
                                                       ProcessID()
                                                                                int;
22
    Description
23
          Gets the ID number assigned to the process.
24
          RequestCount
25
```

Init 2 RequestCount [C#] public {get;} int public: [C++]property int get RequestCount(); Public ReadOnly Property RequestCount Integer [VB] As [JScript] public function get RequestCount() int; 7 Description Gets the number of start requests for the process. 9 ShutdownReason 10 Init 11 12 ProcessShutdownReason [C#] public ShutdownReason {get;} 13 [C++] public: property ProcessShutdownReason get ShutdownReason(); [VB] Public ReadOnly Property ShutdownReason As ProcessShutdownReason 15 [JScript] public function get ShutdownReason(): ProcessShutdownReason; 16 17 Description 18 Gets a value that indicates why the process shut down. 19 StartTime 20 Init 21 22 public DateTime StartTime [C#] {get;} 23 [C++]public: DateTime get StartTime(); property 24 Public ReadOnly **Property** StartTime DateTime [VB] As

[JScript] public function StartTime() DateTime; get 2 Description 3 Gets the time at which the process started. Status Init 7 ProcessStatus [C#] public Status {get;} 8 [C++]public: ProcessStatus get Status(); property [VB] Public ReadOnly Status **ProcessStatus** Property As 10 [JScript] public function Status() ProcessStatus; get 11 12 Description 13 Gets the current status of the process. 14 SetAll 15 16 [C#] public void SetAll(DateTime startTime, TimeSpan age, int processID, int requestCount, ProcessStatus status, ProcessShutdownReason shutdownReason, int 18 peakMemoryUsed); 19 [C++] public: void SetAll(DateTime startTime, TimeSpan age, int processID, int 20 requestCount, ProcessStatus status, ProcessShutdownReason shutdownReason, int 21 peakMemoryUsed); 22 [VB] Public Sub SetAll(ByVal startTime As DateTime, ByVal age As TimeSpan, 23 ByVal processID As Integer, ByVal requestCount As Integer, ByVal status As 24 ProcessStatus, ByVal shutdownReason As ProcessShutdownReason, ByVal

peakMemoryUsed As Integer)

[JScript] public function SetAll(startTime : DateTime, age : TimeSpan, processID

: int, requestCount : int, status : ProcessStatus, shutdownReason :

ProcessShutdownReason, peakMemoryUsed : int);

#### Description

3

5

7

10

11

12

13

14

15

16

17

18

19

20

21

23

24

25

Sets internal information indicating the status of the process. A System.DateTime that indicates the time at which the process started. A System.TimeSpan that indicates the time elapsed since the process started. The ID number assigned to the process. The number of start requests for the process. One of the System.Web.ProcessStatus values that indicates the time elapsed since the process started. One of the System.Web.ProcessShutdownReason values. The maximum memory used (in bytes).

ProcessModelInfo class (System.Web)

**ToString** 

ProcessModelInfo

Example Syntax:

**ToString** 

**GetCurrentProcessInfo** 

[C#] public static ProcessInfo GetCurrentProcessInfo();
[C++] public: static ProcessInfo\* GetCurrentProcessInfo();
[VB] Public Shared Function GetCurrentProcessInfo() As ProcessInfo
[JScript] public static function GetCurrentProcessInfo(): ProcessInfo;

GetHistory

[C#] public ProcessInfo[] GetHistory(int numRecords); static ProcessInfo\* GetHistory(int [C++]public: static numRecords) [];[VB] Public Shared Function GetHistory(ByVal numRecords As Integer) As ProcessInfo() [JScript] public static function GetHistory(numRecords : int) : ProcessInfo[]; ProcessShutdownReason enumeration (System.Web) 7 **ToString** 8 9 10 Description 11 Provides enumerated values that indicate why a process has shut down. 12 **ToString** 13 14 [C#] public ProcessShutdownReason IdleTimeout; const 15 ProcessShutdownReason [C++]public: IdleTimeout; const 16 IdleTimeout [VB] Public Const As ProcessShutdownReason 17 [JScript] public IdleTimeout ProcessShutdownReason; var 18 19 Description 20 Indicates that the process exceeded the allowable idle time. 21 **ToString** 22 23 public ProcessShutdownReason MemoryLimitExceeded; [C#] const 24 public: ProcessShutdownReason MemoryLimitExceeded; const

```
[VB]
           Public
                   Const
                            MemoryLimitExceeded As ProcessShutdownReason
                                                         ProcessShutdownReason;
    [JScript]
              public
                             MemoryLimitExceeded
                       var
3
   Description
          Indicates that the process exceeded the per-process memory limit.
5
          ToString
6
7
    [C#]
                                            ProcessShutdownReason
                public
                               const
                                                                            None;
   [C++]
                 public:
                                             ProcessShutdownReason
                                                                            None;
                                const
    [VB]
               Public
                                      None
                                                  As
                                                          ProcessShutdownReason
                           Const
10
                                                          ProcessShutdownReason;
    [JScript]
                  public
                                       None
                                                   :
                              var
11
12
    Description
13
          Indicates that the process has not shut down.
14
          ToString
15
16
                                         ProcessShutdownReason
                                                                       PingFailed;
    [C#]
               public
                            const
    [C++]
                public:
                                         ProcessShutdownReason
                                                                       PingFailed;
                             const
18
    [VB]
              Public
                                   PingFailed
                                                          ProcessShutdownReason
                         Const
                                                   As
    [JScript]
                 public
                                    PingFailed
                                                          ProcessShutdownReason;
                            var
20
21
    Description
22
23
          ToString
24
25
```

ProcessShutdownReason RequestQueueLimit; public [C#] const RequestQueueLimit; [C++]public: ProcessShutdownReason const ProcessShutdownReason [VB] Public Const RequestQueueLimit As RequestQueueLimit ProcessShutdownReason; [JScript] public var

## Description

1

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Indicates that the request for a process exceeded the allowable number of processes in the queue.

**ToString** 

[C#] public ProcessShutdownReason RequestsLimit; const [C++]public: const ProcessShutdownReason RequestsLimit; [VB] Public RequestsLimit ProcessShutdownReason Const As ProcessShutdownReason; [JScript] public RequestsLimit var

# Description

Indicates that the request for the process exceeded the allowable number of processes.

**ToString** 

ProcessShutdownReason [C#] public const Timeout; public: ProcessShutdownReason [C++]Timeout; const Public ProcessShutdownReason [VB] Const **Timeout** As [JScript] ProcessShutdownReason; public Timeout : var

1									
2	Description								
3	Indicates that the process timed out.								
4	To	String							
5									
6	[C#]	public	const	Proc	essShute	lownReason	Unexpected;		
7	[C++]	public:	const	Proc	essShut	downReason	Unexpected;		
8	[VB]	Public	Const	Unexpect	ed A	As Proces	sShutdownReason		
9	[JScript]	public	var	Unexpec	eted	: Process	ShutdownReason;		
10									
11	Descripti	on							
12	In	dicates that	the process	shut down	unexpe	ctedly.			
13	ProcessStatus enumeration (System.Web)								
14	То	String							
15									
16									
17	Descripti	on							
18	Pr	ovides enun	nerated valu	ues that ind	licate the	e current status	s of a process.		
19	To	String							
20									
21	[C#]	publ	ic	const		ProcessStatus	Alive;		
22	[C++]	pub	lic:	const		ProcessStatu	s Alive;		
23	[VB]	Public	Co	nst	Alive	As	ProcessStatus		
24	[JScript]	pub	olic	var	Alive	:	ProcessStatus;		
25									

	1									
	2	Descriptio	n							
	3	Indicates that the process is running.								
	4	ToS	String							
	5	l i								
	6	[C#]	public	const	Process	Status				
	7	[C++]	public:	cons	st Process	Status				
	8	[VB]	Public	Const	ShutDown	As				
	9	[JScript]	public	var	ShutDown	:				
31, 21, 21,	10									
	11	Descriptio	n							
THE THE	12	Ind	shut down.							
	13	ToS	String							
	14									
	15	[C#]	public	const	ProcessSta	tus				
	16	[C++]	public:	const	ProcessSta	atus				
	17	[VB]	Public	Const	ShuttingDown	As				
	18	[JScript]	public	var	ShuttingDown	:				
	19									
	20	Descriptio								
	21			process has l	begun to shut down	1.				
	22	ToS	String							
	23									
	24	[C#]	public	const	ProcessS					
	25	[C++]	public:	cons	t Process	Status				

Terminated;

Terminated;

ShutDown;

ShutDown;

ProcessStatus

ProcessStatus;

ShuttingDown;

ShuttingDown;

ProcessStatus

ProcessStatus;

[VB] Public Const Terminated As ProcessStatus

[JScript] public var Terminated : ProcessStatus;

Description

Indicates that the process has terminated.

TraceContext class (System.Web)

**ToString** 

## Description

3

5

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Captures and presents execution details about a Web request. This class cannot be inherited.

You can use this class to append messages to specific trace categories. For example, if you are creating an instance of the **System.Web.UI.WebControls.Calendar** class for your application, you might include the trace message "Starting To Render" in a rendering procedure, and "Firing OnChange Event" in an event handling procedure.

TraceContext

Example Syntax:

**ToString** 

public TraceContext(HttpContext [C#] context); public: TraceContext(HttpContext\* [C++]context); [VB] Public Sub New(ByVal HttpContext) context As [JScript] public function TraceContext(context HttpContext);

#### Description

Initializes a new instance of the **System.Web.TraceContext** class. An **System.Web.HttpContext** that contains information about the current Web request.

IsEnabled

**ToString** 

[C#] public bool IsEnabled {get; set;}
[C++] public: \_\_property bool get\_IsEnabled();public: \_\_property void set\_IsEnabled(bool);

[VB] Public Property IsEnabled As Boolean [JScript] public function get IsEnabled() : Boolean; public function set IsEnabled(Boolean);

# Description

Indicates whether tracing is enabled for the current Web request.

Use this flag to check whether your page or application should output tracing information before it writes anything to the trace log. You can set this property to **true** for a page by including a trace="true" attribute in the directive. To set the property to **true** for an entire application, set it in the application's Web.config file.

MS1-863US.APP

TraceMode

**ToString** 

[C#] public TraceMode TraceMode {get; set;}
[C++] public: \_\_property TraceMode get\_TraceMode();public: \_\_property void
set\_TraceMode(TraceMode);

[VB] Public Property TraceMode As TraceMode [JScript] public function get TraceMode(): TraceMode; public function set TraceMode(TraceMode);

#### Description

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets the sorted order in which trace messages should be output to a requesting browser.

Trace messages can be sorted in the order they were processed, or alphabetically by user-defined category.

Warn

[C#] public void Warn(string message); Warn(String\* [C++]public: void message); [VB] Public Sub Warn(ByVal message As String) [JScript] public function Warn(message: String); Writes trace information to the trace log. Unlike System.Web.TraceContext.Write(System.String) warnings appear in the log red as text.

#### Description

Writes a trace message to the trace log. All warnings appear as in the log as red text. The trace message to write to the log.

Warn

public

2

3

[C#]

[C++]public:

Warn(String\* void

Warn(string

void

category,

category,

String\*

string

message);

message);

[VB] Public Sub Warn(ByVal category As String, ByVal message As String)

[JScript] public function Warn(category : String, message : String); Writes trace

information to the trace log. All warnings appear in the log as red text.

8

9

6

7

10 11

12

13 14

15 16

17

18 19

20

21 22

23

24 25 Description

Writes trace information to the trace log, including any user-defined categories and trace messages. All warnings appear in the log as red text. The trace category that receives the message. The trace message to write to the log.

Warn

[C#] public void Warn(string category, string message, Exception errorInfo); [C++] public: void Warn(String\* category, String\* message, Exception\* errorInfo);

[VB] Public Sub Warn(ByVal category As String, ByVal message As String, ByVal errorInfo Exception) As [JScript] public function Warn(category : String, message : String, errorInfo : Exception);

Description

Writes trace information to the trace log, including any user-defined categories, trace messages, and error information. All warnings appear in the log as red text. The trace category that receives the message. The trace message to write to the log. An **System.Exception** that contains information about the error.

Write

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] public void Write(string message); [C++]public: void Write(String\* message); [VB] Public Sub Write(ByVal As String) message [JScript] public function Write(message: String); Writes trace information to the trace log.

#### Description

Writes a trace message to the trace log. The trace message to write to the log.

Write

[C#] public void Write(string string category, message); [C++]public: void Write(String\* category, String\* message); [VB] Public Sub Write(ByVal category As String, ByVal message As String) [JScript] public function Write(category : String, message : String); Writes trace information to the trace log.

#### Description

Writes trace information to the trace log, including a message and any userdefined categories. The trace category that receives the message. The trace message to write to the log.

2

5

7

9 10

11

13 14 15

16 17

18

19 20

21

22 23

24

25

Write

[C#] public void Write(string category, string message, Exception errorInfo); [C++] public: void Write(String\* category, String\* message, Exception\* errorInfo);

[VB] Public Sub Write(ByVal category As String, ByVal message As String, ByVal errorInfo As Exception)

[JScript] public function Write(category : String, message : String, errorInfo :

Exception);

# Description

Writes trace information to the trace log, including any user-defined categories, trace messages, and error information. The trace category that receives the message. The trace message to write to the log. An **System.Exception** that contains information about the error.

TraceMode enumeration (System.Web)

Write

# Description

Specifies in what order trace messages are emitted into the HTML output of a page.

Write

[C#] public const TraceMode Default;

[C++]	public:	const	11	raceMode	Default;
[VB]	Public	Const	Default	As	TraceMode
[JScript]	public	var	Default	:	TraceMode;

# Description

Specifies the default value of the  $\mathbf{TraceMode}$  enumeration, which is  $\mathbf{SortByTime}$ .

### System.Web.Caching

#### Description

The System.Web.Caching namespace provides classes for caching frequently used resources on the server. This includes ASP.NET pages, web services, and user controls. Additionally, a cache dictionary is available for you to store frequently used resources, such as hashtables and other data structures.

Cache class (System.Web.Caching)

#### Description

Implements the cache for a Web application.

One instance of this class is created per application domain, and it remains valid as long as the application domain remains active. Information about an instance of this class is available through the **Cache** property of the **System.Web.HttpContext** object. This class cannot be inherited.

lee@hayes pik 509+324-9256 293 MS1-863US.APP

4	
5	
6	
7	,
8	
9	,
10	)
11	
12	!
13	,
14	ı
15	5
16	ó
17	7
18	3
19	)

[C#] public static readonly DateTime NoAbsoluteExpiration;
[C++] public: static DateTime NoAbsoluteExpiration;
[VB] Public Shared ReadOnly NoAbsoluteExpiration As DateTime
[JScript] public static var NoAbsoluteExpiration : DateTime;
Description
Used in the absoluteExpiration parameter in an
System.Web.Caching.Cache.Insert(System.String,System.Object) method call
to indicate the item should never expire. This field is read-only.
When used, this field sets the absoluteExpiration parameter equal to
System.DateTime.MaxValue, which is a constant representing the largest
possible <b>DateTime</b> value, 12/31/9999 11:59:59 PM.
[C#] public static readonly TimeSpan NoSlidingExpiration;
[C++] public: static TimeSpan NoSlidingExpiration;
[VB] Public Shared ReadOnly NoSlidingExpiration As TimeSpan
[JScript] public static var NoSlidingExpiration : TimeSpan;
Description
Used as the slidingExpiration parameter in an
System.Web.Caching.Cache.Insert(System.String,System.Object) method cal
to disable sliding expirations. This field is read-only.

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Item

When used, this field sets the slidingExpiration parameter equal to the System.TimeSpan.Zero field, which has a constant value of zero. The cached item then expires in accordance with the absoluteExpiration parameter. Constructors: Cache Example Syntax: [C#] public Cache(); [C++] public: Cache(); [VB] Public Sub New() [JScript] public function Cache(); Properties: Count [C#] public int Count {get;} [C++] public: property int get Count(); [VB] Public ReadOnly Property Count As Integer [JScript] public function get Count(): int; Description Gets the number of items stored in the cache. This property can be useful when monitoring your application's performance or when using ASP.NET tracing functionality.

[C#] public object this[string key] {get; set;}
[C++] public: \_\_property Object\* get\_Item(String\* key);public: \_\_property void
set\_Item(String\* key, Object\*);
[VB] Public Default Property Item(ByVal key As String) As Object
[JScript] returnValue = CacheObject.Item(key);CacheObject.Item(key) =
returnValue;

Description

Gets or sets the cache item at the specified key.

You can use this property to retrieve the value of a specified cache item, or to add an item and a key for it to the cache. A **System.String** object that represents the key for the cache item.

Methods:

Add

[C#] public object Add(string key, object value, CacheDependency dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration, CacheItemPriority priority, CacheItemRemovedCallback onRemoveCallback);
[C++] public: Object\* Add(String\* key, Object\* value, CacheDependency\* dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration,
CacheItemPriority priority, CacheItemRemovedCallback\* onRemoveCallback);
[VB] Public Function Add(ByVal key As String, ByVal value As Object, ByVal dependencies As CacheDependency, ByVal absoluteExpiration As DateTime,
ByVal slidingExpiration As TimeSpan, ByVal priority As CacheItemPriority,

ByVal onRemoveCallback As CacheItemRemovedCallback) As Object

[JScript] public function Add(key: String, value: Object, dependencies:

CacheDependency, absoluteExpiration: DateTime, slidingExpiration: TimeSpan, priority: CacheItemPriority, onRemoveCallback: CacheItemRemovedCallback):

Object;

#### Description

Adds the specified item to the **System.Web.Caching.Cache** object with dependencies, expiration and priority policies, and a delegate you can use to notify your application when the inserted item is removed from the **Cache**.

Return Value: The System.Object item added to the Cache. The cache key used to reference the item. The item to be added to the cache. The file or cache key dependencies for the item. When any dependency changes, the object becomes invalid and is removed from the cache. If there are no dependencies, this paramter contains null. The time at which the added object expires and is removed from the cache. The interval between the time the added object was last accessed and when that object expires. If this value is the equivalent of 20 minutes, the object expires and is removed from the cache 20 minutes after it is last accessed. The relative cost of the object, as expressed by the System.Web.Caching.CacheItemPriority enumeration. The cache uses this value when it evicts objects; objects with a lower cost are removed from the cache before objects with a higher cost. A delegate that, if provided, is called when an object is removed from the cache. You can use this to notify applications when their objects are deleted from the cache.

Get

1

16 17

15

19

20

18

21

22

23

24 25 [C#] public object Get(string key);

[C++] public: Object\* Get(String\* key);

[VB] Public Function Get(ByVal key As String) As Object

[JScript] public function Get(key: String): Object;

## Description

Retrieves the specified item from the **System.Web.Caching.Cache** object.

Return Value: The retrieved cache item, or **null** if the key is not found. The identifier for the cache item to retrieve.

GetEnumerator

[C#] public IDictionaryEnumerator GetEnumerator();

[C++] public: IDictionaryEnumerator\* GetEnumerator();

[VB] Public Function GetEnumerator() As IDictionaryEnumerator

[JScript] public function GetEnumerator(): IDictionaryEnumerator;

#### Description

Retrieves a dictionary enumerator used to iterate through the key settings and their values contained in the cache.

Return Value: An enumerator to iterate through the System. Web. Caching. Cache object.

Items can be added or removed from the cache while this method is enumerating through the items.

Insert

1	
2	[C#] public void Insert(string key, object value);
3	[C++] public: void Insert(String* key, Object* value);
4	[VB] Public Sub Insert(ByVal key As String, ByVal value As Object)
5	[JScript] public function Insert(key: String, value: Object); Inserts an item into
6	the System.Web.Caching.Cache object.
7	
8	Description
9	Inserts an item into the System.Web.Caching.Cache object with a cache
10	key to reference its location and using default values provided by the
11	System.Web.Caching.CacheItemPriority and
12	System.Web.Caching.CacheItemPriorityDecay enumerations. The cache key
13	used to reference the item. The object to be inserted into the cache.
14	Insert
15	
16	[C#] public void Insert(string key, object value, CacheDependency dependencies);
17	[C++] public: void Insert(String* key, Object* value, CacheDependency*
18	dependencies);
19	[VB] Public Sub Insert(ByVal key As String, ByVal value As Object, ByVal
20	dependencies As CacheDependency)
21	[JScript] public function Insert(key: String, value: Object, dependencies:
22	CacheDependency);
23	
24	Description
25	

Inserts an object into the **System.Web.Caching.Cache** that has file or key dependencies. The cache key used to identify the item. The object to be inserted in the cache. The file or cache key dependencies for the inserted object. When any dependency changes, the object becomes invalid and is removed from the cache. If there are no dependencies, this parameter contains **null**.

Insert

[C#] public void Insert(string key, object value, CacheDependency dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration);

[C++] public: void Insert(String\* key, Object\* value, CacheDependency\* dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration); [VB] Public Sub Insert(ByVal key As String, ByVal value As Object, ByVal dependencies As CacheDependency, ByVal absoluteExpiration As DateTime, ByVal slidingExpiration As TimeSpan)

[JScript] public function Insert(key: String, value: Object, dependencies: CacheDependency, absoluteExpiration: DateTime, slidingExpiration: TimeSpan);

Description

Inserts an object into the **System.Web.Caching.Cache** with dependencies and expiration policies.

If the *slidingExpiration* parameter is set to **NoSlidingExpiration**, sliding expiration is disabled. If you set the *slidingExpiration* parameter to greater than **System.TimeSpan.Zero**, the *absoluteExpiration* parameter is set to **System.DateTime.Now** plus the value contained in the *slidingExpiration* 

parameter. If the item is requested from the cache before the amount of time specified by the *absoluteExpiration* parameter, the item will be placed in the cache again, and *absoluteExpiration* will again be set to **DateTime.Now** plus the value contained in the *slidingExpiration* parameter. If the item is not requested from the cache before the date in the *absoluteExpiration* parameter, the item is removed from the cache. The cache key used to reference the object. The object to be inserted in the cache. The file or cache key dependencies for the inserted object. When any dependency changes, the object becomes invalid and is removed from the cache. If there are no dependencies, this parameter contains **null**. The time at which the inserted object expires and is removed from the cache. The interval between the time the inserted object is last accessed and when that object expires. If this value is the equivalent of 20 minutes, the object will expire and be removed from the cache 20 minutes after it was last accessed.

Insert

[C#] public void Insert(string key, object value, CacheDependency dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration, CacheItemPriority priority, CacheItemRemovedCallback onRemoveCallback);
[C++] public: void Insert(String\* key, Object\* value, CacheDependency\* dependencies, DateTime absoluteExpiration, TimeSpan slidingExpiration, CacheItemPriority priority, CacheItemRemovedCallback\* onRemoveCallback);
[VB] Public Sub Insert(ByVal & String, ByVal value As Object, ByVal dependencies As CacheDependency, ByVal absoluteExpiration As DateTime, ByVal slidingExpiration As TimeSpan, ByVal priority As CacheItemPriority, ByVal onRemoveCallback As CacheItemRemovedCallback)

[JScript] public function Insert(key: String, value: Object, dependencies: CacheDependency, absoluteExpiration: DateTime, slidingExpiration: TimeSpan, priority: CacheItemPriority, onRemoveCallback: CacheItemRemovedCallback);

#### Description

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Inserts an object into the System. Web. Caching. Cache object with dependencies, expiration and priority policies, and a delegate you can use to notify your application when the inserted item is removed from the Cache. The cache key used to reference the object. The object to be inserted in the cache. The file or cache key dependencies for the item. When any dependency changes, the object becomes invalid and is removed from the cache. If there are no dependencies, this parameter contains **null**. The time at which the inserted object expires and is removed from the cache. The interval between the time the inserted object was last accessed and when that object expires. If this value is the equivalent of 20 minutes, the object will expire and be removed from the cache 20 minutes after it was last accessed. The cost of the object relative to other items stored in the cache, as expressed by the System. Web. Caching. Cacheltem Priority enumeration. This value is used by the cache when it evicts objects; objects with a lower cost are removed from the cache before objects with a higher cost. A delegate that, if provided, will be called when an object is removed from the cache. You can use this to notify applications when their objects are deleted from the cache.

Remove

[C#] public object Remove(string key);

[C++] public: Object\* Remove(String\* key);

1	[VB] Public Function Remove(ByVal key As String) As Object
2	[JScript] public function Remove(key : String) : Object;
3	
4	Description
5	Removes the specified item from the application's
6	System.Web.Caching.Cache object.
7	Return Value: The item removed from the Cache. If the value in the key
8	parameter is not found, returns null. A System.String identifier for the cache
9	item to remove.
10	IEnumerable.GetEnumerator
11	
12	[C#] IEnumerator IEnumerable.GetEnumerator();
13	[C++] IEnumerator* IEnumerable::GetEnumerator();
14	[VB] Function GetEnumerator() As IEnumerator Implements
15	IEnumerable.GetEnumerator
16	[JScript] function IEnumerable.GetEnumerator(): IEnumerator;
17	CacheDependency class (System.Web.Caching)
18	ToString
19	
20	
21	Description
22	Tracks cache dependencies, which can be files, directories, or keys to or
- 1	

ys to other objects in your application's System. Web. Caching. Cache . This class cannot be

inherited. 24

25

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

1

3

You can add items with dependencies to your application's cache with the System.Web.Caching.Cache.Add(System.String,System.Object,System.Web. Caching.CacheDependency,System.DateTime,System.TimeSpan,System.Web. Caching.CacheItemPriority,System.Web.Caching.CacheItemRemovedCallback) and System.Web.Caching.Cache.Insert(System.String,System.Object) methods.

CacheDependency

Example Syntax:

**ToString** 

[C#] public CacheDependency(string filename);

[C++] public: CacheDependency(String\* filename);

[VB] Public Sub New(ByVal filename As String)

[JScript] public function CacheDependency(filename : String); Initializes a new instance of the **System.Web.Caching.CacheDependency** class.

Description

Initializes a new instance of the **System.Web.Caching.CacheDependency** class that monitors a file or directory for changes. The path to a file or directory that the cached object is dependent upon. When this resource changes, the cached object becomes obsolete and is removed from the cache.

CacheDependency

Example Syntax:

	3
	4
	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
	15
	16
	17
	18
	19
	20
	21

[C#] public CacheDependency(string[] filenames);
[C++] public: CacheDependency(String* filenamesgc[]);
[VB] Public Sub New(ByVal filenames() As String)
[JScript] public function CacheDependency(filenames : String[])

Description

Initializes a new instance of the **System.Web.Caching.CacheDependency** class that monitors an array of file paths (to files or directories) for changes.

If any of the files or directories in the array were to change or be removed from the array, the cached item becomes obsolete and is removed from the application's **System.Web.Caching.Cache** object. An array of file paths (to files or directories) that the cached object is dependent upon. When any of these resources change, the cached object becomes obsolete and is removed from the cache.

CacheDependency

Example Syntax:

ToString

[C#] public CacheDependency(string filename, DateTime start);[C++] public: CacheDependency(String\* filename, DateTime start);[VB] Public Sub New(ByVal filename As String, ByVal start As DateTime)[JScript] public function CacheDependency(filename : String, start : DateTime);

Description

23

2

3

4

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the System. Web. Caching. Cache Dependency class that monitors a file or directory for changes and indicates when change tracking is to begin. The path to a file or directory that the cached object is dependent upon. When this resource changes, the cached object becomes obsolete and is removed from the cache. The time when change tracking begins.

CacheDependency

Example Syntax:

**ToString** 

[C#] public CacheDependency(string[] filenames, DateTime start); [C++] public: CacheDependency(String\* filenames gc[], DateTime start); [VB] Public Sub New(ByVal filenames() As String, ByVal start As DateTime) [JScript] public function CacheDependency(filenames : String[], start : DateTime);

### Description

Initializes a new instance of the System. Web. Caching. Cache Dependency class that monitors an array of file paths (to files or directories) for changes and specifies a time when change monitoring begins. An array of file paths (to files or directories) that the cached object is dependent upon. When any of these resources change, the cached object becomes obsolete and is removed from the cache. The time when change tracking begins.

CacheDependency

Example Syntax:

5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

16

17

18

19

20

21

22

23

24

3

[C#] public CacheDependency(string[] filenames, string[] cachekeys);
[C++] public: CacheDependency(String\* filenames \_\_gc[], String\* cachekeys
\_\_gc[]);
[VB] Public Sub New(ByVal filenames() As String, ByVal cachekeys() As String)
[JScript] public function CacheDependency(filenames : String[], cachekeys :
String[]);

### Description

Initializes a new instance of the **System.Web.Caching.CacheDependency** class that monitors an array of file paths (to files or directories), an array of cache keys, or both for changes. An array of file paths (to files or directories) that the cached object is dependent upon. When any of these resources change, the cached object becomes obsolete and is removed from the cache. An array of cache keys that the new object monitors for changes. When any of these cache keys change, the cached object associated with this dependency object becomes obsolete and is removed from the cache.

CacheDependency

Example Syntax:

**ToString** 

[C#] public CacheDependency(string[] filenames, string[] cachekeys,

CacheDependency dependency);

[C++] public: CacheDependency(String\* filenames \_\_gc[], String\* cachekeys
\_\_gc[], CacheDependency\* dependency);

[VB] Public Sub New(ByVal filenames() As String, ByVal cachekeys() As String, ByVal dependency As CacheDependency)

[JScript] public function CacheDependency(filenames : String[], cachekeys :

String[], dependency : CacheDependency);

### Description

Initializes a new instance of the **System.Web.Caching.CacheDependency** class that monitors an array of file paths (to files or directories), an array of cache keys, or both for changes. It also makes itself dependent upon a separate instance of the **CacheDependency** class. An array of file paths (to files or directories) that the cached object is dependent upon. When any of these resources change, the cached object becomes obsolete and is removed from the cache. An array of cache keys that the new object monitors for changes. When any of these cache keys change, the cached object associated with this dependency object becomes obsolete and is removed from the cache. Another instance of the

CacheDependency

Example Syntax:

**ToString** 

[C#] public CacheDependency(string[] filenames, string[] cachekeys, DateTime start);

CacheDependency class that this instance is dependent upon.

[C++] public: CacheDependency(String\* filenames \_\_gc[], String\* cachekeys \_\_gc[], DateTime start);

[VB] Public Sub New(ByVal filenames() As String, ByVal cachekeys() As String,

dies. Jun. des des

ByVal start As DateTime)

[JScript] public function CacheDependency(filenames : String[], cachekeys :

String[], start : DateTime);

Description

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the **System.Web.Caching.CacheDependency** class that monitors an array of file paths (to files or directories), an array of cache keys, or both for changes. An array of file paths (to files or directories) that the cached object is dependent upon. When any of these resources change, the cached object becomes obsolete and is removed from the cache. An array of cache keys that the new object monitors for changes. When any of these cache keys change, the cached object associated with this dependency object becomes obsolete and is removed from the cache. The time when change tracking begins.

CacheDependency

Example Syntax:

**ToString** 

[C#] public CacheDependency(string[] filenames, string[] cachekeys,

CacheDependency dependency, DateTime start);

[C++] public: CacheDependency(String\* filenames \_\_gc[], String\* cachekeys

\_\_gc[], CacheDependency\* dependency, DateTime start);

[VB] Public Sub New(ByVal filenames() As String, ByVal cachekeys() As String.

ByVal dependency As CacheDependency, ByVal start As DateTime)

[JScript] public function CacheDependency(filenames : String[], cachekeys :

String[], dependency: CacheDependency, start: DateTime);

### Description

Initializes a new instance of the **System.Web.Caching.CacheDependency** class that monitors an array of file paths (to files or directories), an array of cache keys, or both for changes. It also makes itself dependent upon another instance of the **CacheDependency** class and a time when the change monitoring begins. An array of file paths (to files or directories) that the cached object is dependent upon. When any of these resources change, the cached object becomes obsolete and is removed from the cache. An array of cache keys that the new object monitors for changes. When any of these cache keys change, the cached object associated with this dependency object becomes obsolete and is removed from the cache. Another instance of the **CacheDependency** class that this instance is dependent upon. The time when change tracking begins.

Dispose

[C#] public void Dispose();

[C++] public: sealed void Dispose();

[VB] NotOverridable Public Sub Dispose()

[JScript] public function Dispose();

Description

Releases the resources used by the

System. Web. Caching. Cache Dependency object.

CacheItemPriority enumeration (System.Web.Caching)

8

9

6

5

2

3

10

12 13

15

14

16 17

18

19

2021

22

23

24

Description

Specifies the relative priority of items stored in the System.Web.Caching.Cache .

When the Web server hosting an ASP.NET application runs low on memory, the **Cache** selectively purges items to free system memory. When an item is added to the cache, you can assign it a relative priority compared to the other items stored in the cache. Items you assign higher priority values to are less likely to be deleted from the cache when the server is processing a large number of requests, while items you assign lower priority values are more likely to be deleted. The default is **Normal**.

**ToString** 

[C#] public const CacheItemPriority AboveNormal;

[C++] public: const CacheItemPriority AboveNormal;

[VB] Public Const AboveNormal As CacheItemPriority

[JScript] public var AboveNormal : CacheItemPriority;

Description

Cache items with this priority level are less likely to be deleted as the server frees system memory than those assigned a **Normal** priority.

**ToString** 

[C#] public const CacheItemPriority BelowNormal;

1	[C++] public: const CacheltemPriority BelowNormal;
2	[VB] Public Const BelowNormal As CacheItemPriority
3	[JScript] public var BelowNormal : CacheItemPriority;
4	
5	Description
6	Cache items with this priority level are more likely to be deleted from the
7	cache as the server frees system memory than items assigned a Normal priority.
8	ToString
9	
10	[C#] public const CacheItemPriority Default;
11	[C++] public: const CacheItemPriority Default;
12	[VB] Public Const Default As CacheItemPriority
13	[JScript] public var Default : CacheItemPriority;
14	
15	Description
16	The default value for a cached item's priority is <b>Normal</b> .
17	ToString
18	
19	[C#] public const CacheItemPriority High;
20	[C++] public: const CacheItemPriority High;
21	[VB] Public Const High As CacheItemPriority
22	[JScript] public var High: CacheItemPriority;
23	
24	Description
25	

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Cache items with this priority level are the least likely to be deleted from the cache as the server frees system memory.

ToString

[C#] public const CacheItemPriority Low;

[C++] public: const CacheItemPriority Low;

[VB] Public Const Low As CacheItemPriority

[JScript] public var Low: CacheItemPriority;

## Description

Cache items with this priority level are the most likely to be deleted from the cache as the server frees system memory.

**ToString** 

[C#] public const CacheItemPriority Normal;

[C++] public: const CacheItemPriority Normal;

[VB] Public Const Normal As CacheItemPriority

[JScript] public var Normal : CacheItemPriority;

# Description

Cache items with this priority level are likely to be deleted from the cache as the server frees system memory only after those items with **Low** or

313

BelowNormal priority. This is the default.

-1111 . 1224 . 1.1.1

1

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

[C#] public const CacheItemPriority NotRemovable;

[C++] public: const CacheItemPriority NotRemovable;

[VB] Public Const NotRemovable As CacheItemPriority

[JScript] public var NotRemovable : CacheItemPriority;

## Description

The cache items with this priority level will not be deleted from the cache as the server frees system memory.

CacheItemRemovedCallback delegate (System.Web.Caching)

**ToString** 

## Description

Defines a callback method for notifying applications when a cached item is removed from the **System.Web.Caching.Cache**. The index location for the item removed from the cache. The **System.Object** item removed from the cache. The reason the item was removed from the cache, as specified by the

System. Web. Caching. Cache I tem Removed Reason enumeration.

CacheItemRemovedReason enumeration (System.Web.Caching)

**ToString** 

22

23

24

25

Description

lee@hayes pilk 509+324-9256 314 MS1-863US,APP

Specifies the reason an item was removed from the 1 System. Web. Caching. Cache. 2 This enumeration works in concert with the 3 System.Web.Caching.CacheItemRemovedCallback delegate to notify your ASP.NET applications when and why an object was removed from the System. Web. Caching. Cache. **ToString** 7 8 [C#] public const CacheItemRemovedReason DependencyChanged; 9 [C++] public: const CacheItemRemovedReason DependencyChanged; 10 [VB] Public Const DependencyChanged As CacheItemRemovedReason 11 [JScript] public var DependencyChanged: CacheItemRemovedReason; 12 13 Description 14 The item is removed from the cache because a file or key dependency 15 changed. 16 **ToString** 17 18 [C#] public const CacheItemRemovedReason Expired; 19 [C++] public: const CacheItemRemovedReason Expired; 20 [VB] Public Const Expired As CacheItemRemovedReason 21 [JScript] public var Expired : CacheItemRemovedReason; 22 23

Description

24

25

The item is removed from the cache because it expired.

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

_				
1	'oS	tr	1n	C
	$\mathbf{v}$		111	. C

[C#] public const CacheItemRemovedReason Removed;

[C++] public: const CacheItemRemovedReason Removed;

[VB] Public Const Removed As CacheItemRemovedReason

[JScript] public var Removed: CacheItemRemovedReason;

## Description

The item is removed fro

# System. Web. Configuration

# Description

The System.Web.Configuration namespace contains classes that are used to set up ASP.NET configuration.

AuthenticationMode enumeration (System.Web.Configuration)

### Description

Provides enumerated values that specify the authentication mode for an application.

[C#] public const AuthenticationMode Forms;

[C++] public: const AuthenticationMode Forms;

1	[VB] Public Const Forms As AuthenticationMode
2	[JScript] public var Forms : AuthenticationMode;
3	
4	Description
5	Specifies ASP.NET forms-based authentication as the default
6	authentication mode.
7	
8	[C#] public const AuthenticationMode None;
9	[C++] public: const AuthenticationMode None;
10	[VB] Public Const None As AuthenticationMode
11	[JScript] public var None: AuthenticationMode;
12	
13	Description
14	Specifies no authentication. Only anonymous users are expected. Also,
15	applications can handle events to provide their own authentication.
16	
17	[C#] public const AuthenticationMode Passport;
18	[C++] public: const AuthenticationMode Passport;
19	[VB] Public Const Passport As AuthenticationMode
20	[JScript] public var Passport : AuthenticationMode;
21	
22	Description
23	Specifies Microsoft Passport authentication as the default authentication
	mode

1	
2	[C#] public const AuthenticationMode Windows;
3	[C++] public: const AuthenticationMode Windows;
4	[VB] Public Const Windows As AuthenticationMode
5	[JScript] public var Windows : AuthenticationMode;
6	
7	Description
8	Specifies Windows authentication as the default authentication mode. Use
9	this mode when using any form of Internet Information Services (IIS)
10	authentication: Basic, Digest, Integrated Windows authentication
11	(NTLM/Kerberos), or certificates.
12	Methods:
13	ClientTargetSectionHandler class (System.Web.Configuration)
14	ToString
15	
16	
17	Description
18	
19	Constructors:
20	ClientTargetSectionHandler
21	Example Syntax:
22	ToString
23	
24	[C#] public ClientTargetSectionHandler();
25	[C++] public: ClientTargetSectionHandler();

1	[VB] Public Sub New()
2	[JScript] public function ClientTargetSectionHandler();
3	Properties:
4	KeyAttributeName
5	ToString
6	
7	[C#] protected override string KeyAttributeName {get;}
8	[C++] protected:property virtual String* get_KeyAttributeName();
9	[VB] Overrides Protected ReadOnly Property KeyAttributeName As String
10	[JScript] protected function get KeyAttributeName(): String;
11	
12	Description
13	Gets the name of the key attribute tag. This property is overidden by
14	derived classes to change the name of the key attribute tag. The default is "key".
15	ValueAttributeName
16	ToString
17	
18	[C#] protected override string ValueAttributeName {get;}
19	[C++] protected:property virtual String* get_ValueAttributeName();
20	[VB] Overrides Protected ReadOnly Property ValueAttributeName As String
21	[JScript] protected function get ValueAttributeName(): String;
22	
23	Description
24	Gets the name of the value tag. This property may be overidden by derived
25	classes to change the name of the value tag. The default is "value"

1	FormsAuthPasswordFormat enumeration (System.Web.Configuration)
2	ToString
3	
4	
5	Description
6	Provides enumerated values that specify the encryption format for storing
7	passwords.
8	ToString
9	
10	[C#] public const FormsAuthPasswordFormat Clear;
11	[C++] public: const FormsAuthPasswordFormat Clear;
12	[VB] Public Const Clear As FormsAuthPasswordFormat
13	[JScript] public var Clear: FormsAuthPasswordFormat;
14	
15	Description
16	Specifies that passwords are not encrypted.
17	ToString
18	
19	[C#] public const FormsAuthPasswordFormat MD5;
20	[C++] public: const FormsAuthPasswordFormat MD5;
21	[VB] Public Const MD5 As FormsAuthPasswordFormat
22	[JScript] public var MD5 : FormsAuthPasswordFormat;
23	
24	Description
25	Specifies that passwords are encrypted using the MD5 hash algorithm.

	1	II
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
1	0	
1	1	
]	12	
1		
	14	
!		
	16 17	

1	ToString
2	
3	[C#] public const FormsAuthPasswordFormat SHA1;
4	[C++] public: const FormsAuthPasswordFormat SHA1;
5	[VB] Public Const SHA1 As FormsAuthPasswordFormat
6	[JScript] public var SHA1 : FormsAuthPasswordFormat;
7	
8	Description
9	Specifies that passwords are encrypted using the SHA1 hash algorithm.
10	FormsProtectionEnum enumeration (System.Web.Configuration)
11	ToString
12	
13	
14	Description
15	Provides enumerated values that are used to specify the forms protection
16	method.
17	ToString
18	
19	[C#] public const FormsProtectionEnum All;
20	[C++] public: const FormsProtectionEnum All;
21	[VB] Public Const All As FormsProtectionEnum
22	[JScript] public var All: FormsProtectionEnum;
23	
24	Description
25	

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Specifies that the application uses both data validation and encryption to protect the cookie. This option uses the configured data validation algorithm (based on the element). Triple-DES (3DES) is used for encryption, if it is available and if the key is at least 48 bytes longs.

**ToString** 

[C#] public const FormsProtectionEnum Encryption;

[C++] public: const FormsProtectionEnum Encryption;

[VB] Public Const Encryption As FormsProtectionEnum

[JScript] public var Encryption: FormsProtectionEnum;

## Description

Specifies that the cookie is encrypted using Triple-DES or DES, but data validation is not performed on the cookie. Cookies used this way might be subject to plain-text security attacks.

**ToString** 

[C#] public const FormsProtectionEnum None;

[C++] public: const FormsProtectionEnum None;

[VB] Public Const None As FormsProtectionEnum

[JScript] public var None : FormsProtectionEnum;

#### Description

Specifies that both encryption and validation are disabled for sites that are using cookies only for personalization and have weaker security requirements.

2

3

5

7

8

9

10

11

12

13

18

19

20

21

24

25

Using cookies in this manner is not recommended; however, it is the least resource-intensive way to enable personalization using the .NET Framework.

**ToString** 

[C#] public const FormsProtectionEnum Validation;

[C++] public: const FormsProtectionEnum Validation;

[VB] Public Const Validation As FormsProtectionEnum

[JScript] public var Validation : FormsProtectionEnum;

# Description

Specifies that the application uses a validation scheme to verify that the contents of an encrypted cookie have not been altered in transit. The cookie is created by concatenating a validation key with the cookie data, computing a Message Authentication Code (MAC), and appending the MAC to the outgoing cookie.

HttpCapabilitiesBase class (System.Web.Configuration)

**ToString** 

# Description

Defines the base class for client browser capabilities.

HttpCapabilitiesBase

Example Syntax:

```
[C#] public HttpCapabilitiesBase();
    [C++] public: HttpCapabilitiesBase();
3
    [VB] Public Sub New()
    [JScript] public function HttpCapabilitiesBase();
5
           Item
6
           ToString
7
8
    [C#] public virtual string this[string key] {get;}
9
    [C++] public: __property virtual String* get_Item(String* key);
10
    [VB] Overridable Public Default ReadOnly Property Item(ByVal key As String)
11
    As String
12
    [JScript] returnValue = HttpCapabilitiesBaseObject.Item(key);
13
14
    Description
15
           Allows access to individual dictionary values. The name of the dictionary
16
    value to retrieve.
17
           GetConfigCapabilities
18
19
    [C#] public static HttpCapabilitiesBase GetConfigCapabilities(string configKey,
    HttpRequest request);
21
    [C++] public: static HttpCapabilitiesBase* GetConfigCapabilities(String*
22
     configKey, HttpRequest* request);
23
    [VB] Public Shared Function GetConfigCapabilities(ByVal configKey As String,
24
     ByVal request As HttpRequest) As HttpCapabilitiesBase
```

	1	[JScript] public static function GetConfigCapabilities(configKey: String, request:
	2	HttpRequest): HttpCapabilitiesBase;
	3	
	4	Description
	5	Returns individual browser capabilities for the current reqest. The name of
	6	the requested browser capability. The current System.Web.HttpContext.Request
	7	•
	8	Init
	9	
3	10	[C#] protected virtual void Init();
a a a a a a a a a a a a a a a a a a a	11	[C++] protected: virtual void Init();
8, 8, 8, 31, 47, 14, 18, 17, 19, 19, 19, 19, 19, 19, 19, 19, 19, 19	12	[VB] Overridable Protected Sub Init()
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	13	
	14	
18 18 18 18 18 18 18 18 18 18 18 18 18 1	15	
# # # # # # # # # # # # # # # # # # #	16	System.Web.Hosting
a.	17	
	18	Description
	19	
	20	AppDomainFactory class (System.Web.Hosting)
	21	
	22	
	23	Description
	24	
	25	Constructors:

	1	AppDomainFactory
	2	Example Syntax:
	3	
	4	[C#] public AppDomainFactory();
	5	[C++] public: AppDomainFactory();
	6	[VB] Public Sub New()
	7	[JScript] public function AppDomainFactory();
	8	Methods:
	9	Create
tion,	10	
dur.	11	[C#] public object Create(string module, string typeName, string appId, string
	12	appPath, string strUrlOfAppOrigin, int iZone);
	13	[C++] public:sealed Object* Create(String* module, String* typeName,
	14	String* appId, String* appPath, String* strUrlOfAppOrigin, int iZone);
	15	[VB] NotOverridable Public Function Create(ByVal module As String, ByVal
	16	typeName As String, ByVal appId As String, ByVal appPath As String, ByVal
	17	strUrlOfAppOrigin As String, ByVal iZone As Integer) As Object
	18	[JScript] public function Create(module : String, typeName : String, appId :
	19	String, appPath: String, strUrlOfAppOrigin: String, iZone: int): Object;
•	20	
	21	Description
	22	
	23	ApplicationHost class (System.Web.Hosting)
	24	ToString

	I	
	2	
	3	Description
	4	
	5	CreateApplicationHost
	6	
	7	[C#] public static object CreateApplicationHost(Type hostType, string virtualDir,
	8	string physicalDir);
	9	[C++] public: static Object* CreateApplicationHost(Type* hostType, String*
# # #:	10	virtualDir, String* physicalDir);
	11	[VB] Public Shared Function CreateApplicationHost(ByVal hostType As Type,
	12	ByVal virtualDir As String, ByVal physicalDir As String) As Object
in Suite	13	[JScript] public static function CreateApplicationHost(hostType: Type, virtualDir
	14	: String, physicalDir : String) : Object;
	15	
	16	Description
	17	
	18	IAppDomainFactory interface (System.Web.Hosting)
	19	ToString
	20	
	21	
	22	Description
	23	
	24	Create

	11	
	1	
	2	[C#] object Create(in string module, in string typeName, in string appId, in string
	3	appPath, in string strUrlOfAppOrigin, in int iZone);
	4	[C++] Object* Create(in String* module,in String* typeName,in String*
	5	appId,in String* appPath,in String* strUrlOfAppOrigin,in int iZone);
	6	[VB] Function Create(ByVal module As String, ByVal typeName As String,
	7	ByVal appId As String, ByVal appPath As String, ByVal strUrlOfAppOrigin As
	8	String, ByVal iZone As Integer) As Object
	9	[JScript] function Create(in module : String, in typeName : String, in appId :
#1 #1	10	String, in appPath: String, in strUrlOfAppOrigin: String, in iZone: int): Object;
Town Town Town York And I I I I I I	11	
	12	Description
	13	
	14	IISAPIRuntime interface (System.Web.Hosting)
	15	Create
	16	
	17	
	18	Description
	19	
	20	DoGCCollect
	21	
	22	[C#] void DoGCCollect();
	23	[C++] void DoGCCollect();
	24	[VB] Sub DoGCCollect()
	25	[JScript] function DoGCCollect();

}}	
1	Description
2	Bescription
3	
4	ProcessRequest
5	
6	[C#] int ProcessRequest(in IntPtr ecb, in int useProcessModel);
7	[C++] int ProcessRequest(_in IntPtr ecb, _in int useProcessModel);
8	[VB] Function ProcessRequest(ByVal ecb As IntPtr, ByVal useProcessModel As
9	Integer) As Integer
 10	[JScript] function ProcessRequest(in ecb : IntPtr, in useProcessModel : int) : int;
11	
12	Description
13	
14	StartProcessing
15	
16	[C#] void StartProcessing();
17	[C++] void StartProcessing();
18	[VB] Sub StartProcessing()
19	[JScript] function StartProcessing();
20	
21	Description
22	
23	StopProcessing
24	
25	[C#] void StopProcessing();

```
[C++] void StopProcessing();
                                                    [VB] Sub StopProcessing()
                                  2
                                                    [JScript] function StopProcessing();
                                  3
                                                    Description
                                  5
                                                                                        ISAPIRuntime class (System.Web.Hosting)
                                                                                         StopProcessing
                                  8
                                  9
The state of the 
                               10
                                                    Description
                              11
                              12
                                                                                         ISAPIRuntime
                               13
                                                                                        Example Syntax:
                               14
                                                                                         StopProcessing
                               15
                               16
                                                    [C#] public ISAPIRuntime();
                               17
                                                    [C++] public: ISAPIRuntime();
                               18
                                                    [VB] Public Sub New()
                               19
                                                    [JScript] public function ISAPIRuntime();
                              20
                                                                                         DoGCCollect
                              21
                              22
                                                    [C#] public void DoGCCollect();
                               23
                                                    [C++] public: __sealed void DoGCCollect();
```

[VB] NotOverridable Public Sub DoGCCollect()

lee@hayes pilc 509+324+9256 330 MS1-863US,APP

StopProcessing

1	
2	[C#] public void StopProcessing();
3	[C++] public:sealed void StopProcessing();
4	[VB] NotOverridable Public Sub StopProcessing()
5	[JScript] public function StopProcessing();
6	
7	Description
8	
9	SimpleWorkerRequest class (System.Web.Hosting)
10	ToString
11	
12	
13	Description
14	
15	SimpleWorkerRequest
16	Example Syntax:
17	ToString
18	
19	[C#] public SimpleWorkerRequest(string page, string query, TextWriter output);
20	[C++] public: SimpleWorkerRequest(String* page, String* query, TextWriter*
21	output);
22	[VB] Public Sub New(ByVal page As String, ByVal query As String, ByVal
23	output As TextWriter)
24	[JScript] public function SimpleWorkerRequest(page : String, query : String,
25	output: TextWriter);
	2   3   4   4   5   6   6   7   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24

11	
1	
2	Description
3	
4	SimpleWorkerRequest
5	Example Syntax:
6	ToString
7	
8	[C#] public SimpleWorkerRequest(string appVirtualDir, string appPhysicalDir,
9	string page, string query, TextWriter output);
10	[C++] public: SimpleWorkerRequest(String* appVirtualDir, String*
11	appPhysicalDir, String* page, String* query, TextWriter* output);
12	[VB] Public Sub New(ByVal appVirtualDir As String, ByVal appPhysicalDir As
13	String, ByVal page As String, ByVal query As String, ByVal output As
14	TextWriter)
15	[JScript] public function SimpleWorkerRequest(appVirtualDir : String,
16	appPhysicalDir: String, page: String, query: String, output: TextWriter);
17	
18	Description
19	
20	Properties:
21	MachineConfigPath
22	ToString
23	
24	[C#] public override string MachineConfigPath {get;}
25	[C++] public:property virtual String* get_MachineConfigPath();
•	

1	[VB] Overrides Public ReadOnly Property MachineConfigPath As String
2	[JScript] public function get MachineConfigPath(): String;
3	
4	Description
5	
6	MachineInstallDirectory
7	ToString
8	
9	[C#] public override string MachineInstallDirectory {get;}
10	[C++] public:property virtual String* get_MachineInstallDirectory();
11	[VB] Overrides Public ReadOnly Property MachineInstallDirectory As String
12	[JScript] public function get MachineInstallDirectory(): String;
13	
14	Description
15	
16	EndOfRequest
17	
18	[C#] public override void EndOfRequest();
19	[C++] public: void EndOfRequest();
20	[VB] Overrides Public Sub EndOfRequest()
21	[JScript] public override function EndOfRequest();
22	
23	Description
24	
25	FlushResponse
	2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24   10   10   10   10   10   10   10   1

1	
2	[C#] public override void FlushResponse(bool finalFlush);
3	[C++] public: void FlushResponse(bool finalFlush);
4	[VB] Overrides Public Sub FlushResponse(ByVal finalFlush As Boolean)
5	[JScript] public override function FlushResponse(finalFlush : Boolean);
6	
7	Description
8	
9	GetAppPath
10	
11	[C#] public override string GetAppPath();
12	[C++] public: String* GetAppPath();
13	[VB] Overrides Public Function GetAppPath() As String
14	[JScript] public override function GetAppPath() : String;
15	
16	Description
17	
18	GetAppPathTranslated
19	
20	[C#] public override string GetAppPathTranslated();
21	[C++] public: String* GetAppPathTranslated();
22	[VB] Overrides Public Function GetAppPathTranslated() As String
23	[JScript] public override function GetAppPathTranslated(): String;
24	
25	Description

40 40 40 40 40 40 40 40 40 40 40 40 40 4		
All the state of t		

3

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

## GetFilePath

[C#] public override string GetFilePath();

[C++] public: String\* GetFilePath();

[VB] Overrides Public Function GetFilePath() As String

[JScript] public override function GetFilePath(): String;

Description

#### GetFilePathTranslated

[C#] public override string GetFilePathTranslated();

[C++] public: String\* GetFilePathTranslated();

[VB] Overrides Public Function GetFilePathTranslated() As String

[JScript] public override function GetFilePathTranslated(): String;

Description

# GetHttpVerbName

[C#] public override string GetHttpVerbName();

[C++] public: String\* GetHttpVerbName();

[VB] Overrides Public Function GetHttpVerbName() As String

[JScript] public override function GetHttpVerbName(): String;

	1	
	2	I
	3	
	4	
	5 6 7	
	6	[
	7	[
	8	]
	9	[
1 31 1 2	10	
	11	1
	12	
	13	
H	14	
	15	
2	16	
	17	
	18	
	19	
	20	    
	21	
	22	

1	
2	Description
3	
4	GetHttpVersion
5	
6	[C#] public override string GetHttpVersion();
7	[C++] public: String* GetHttpVersion();
8	[VB] Overrides Public Function GetHttpVersion() As String
9	[JScript] public override function GetHttpVersion() : String;
10	
11	Description
12	
13	GetLocalAddress
14	
15	[C#] public override string GetLocalAddress();
16	[C++] public: String* GetLocalAddress();
17	[VB] Overrides Public Function GetLocalAddress() As String
18	[JScript] public override function GetLocalAddress() : String;
19	
20	Description
21	
22	GetLocalPort
23	
24	[C#] public override int GetLocalPort();
25	[C++] public: int GetLocalPort();

	1	[V
	2	[J:
	3	
	4	$D\epsilon$
	5	
	6	
	7	
	8	[C
	9	[C
: 2:	10	[/
· · · · · · · · · · · · · · · · · · ·	11	[J
	12	
	13	D
; n <sub>j</sub>	14	
` \  : <b> </b>	15	
	16	
: 2:	17	[(
	18	[(
	19	[ [C
	20	∥ [J
	21	

[VB] Overrides Public Function GetLocalPort() As Integer
[JScript] public override function GetLocalPort(): int;
Description
GetPathInfo
[C#] public override string GetPathInfo();
[C++] public: String* GetPathInfo();
[VB] Overrides Public Function GetPathInfo() As String
[JScript] public override function GetPathInfo(): String;
Description
GetQueryString
[C#] public override string GetQueryString();
[C++] public: String* GetQueryString();
[VB] Overrides Public Function GetQueryString() As String
[JScript] public override function GetQueryString(): String;
Description
GetRawUrl

	1	
	2	[C#] public override string GetRawUrl();
	3	[C++] public: String* GetRawUrl();
	4	[VB] Overrides Public Function GetRawUrl() As String
	5	[JScript] public override function GetRawUrl(): String;
	6	
	7	Description
	8	
	9	GetRemoteAddress
	10	
	11	[C#] public override string GetRemoteAddress();
	12	[C++] public: String* GetRemoteAddress();
	13	[VB] Overrides Public Function GetRemoteAddress() As String
!	14	[JScript] public override function GetRemoteAddress() : String;
	15	
	16	Description
	17	
	18	GetRemotePort
	19	
	20	[C#] public override int GetRemotePort();
	21	[C++] public: int GetRemotePort();
	22	[VB] Overrides Public Function GetRemotePort() As Integer
	23	[JScript] public override function GetRemotePort(): int;
	24	
	25	Description

1	
2	GetServerVariable
3	
4	[C#] public override string GetServerVariable(string name);
5	[C++] public: String* GetServerVariable(String* name);
6	[VB] Overrides Public Function GetServerVariable(ByVal name As String) As
7	String
8	[JScript] public override function GetServerVariable(name : String) : String;
9	
10	Description
11	
12	GetUriPath
13	
14	[C#] public override string GetUriPath();
15	[C++] public: String* GetUriPath();
16	[VB] Overrides Public Function GetUriPath() As String
17	[JScript] public override function GetUriPath(): String;
18	
19	Description
20	
21	GetUserToken
22	
23	[C#] public override IntPtr GetUserToken();
24	[C++] public: IntPtr GetUserToken();
25	[VB] Overrides Public Function GetUserToken() As IntPtr

[JScript] public override function GetUserToken(): IntPtr; 2 Description 3 4 MapPath 5 6 [C#] public override string MapPath(string path); 7 [C++] public: String\* MapPath(String\* path); 8 [VB] Overrides Public Function MapPath(ByVal path As String) As String 9 [JScript] public override function MapPath(path : String) : String; 10 11 Description 12 13 SendKnownResponseHeader 14 15 [C#] public override void SendKnownResponseHeader(int index, string value); 16 [C++] public: void SendKnownResponseHeader(int index, String\* value); 17 [VB] Overrides Public Sub SendKnownResponseHeader(ByVal index As Integer, 18 ByVal value As String) 19 [JScript] public override function SendKnownResponseHeader(index: int, value: 20 String); 21 22 Description 23 24 SendResponseFromFile 25

1	
2	[C#] public override void SendResponseFromFile(IntPtr handle, long offset, long
3	length);
4	[C++] public: void SendResponseFromFile(IntPtr handle,int64 offset,int64
5	length);
6	[VB] Overrides Public Sub SendResponseFromFile(ByVal handle As IntPtr,
7	ByVal offset As Long, ByVal length As Long)
8	[JScript] public override function SendResponseFromFile(handle: IntPtr, offset:
9	long, length: long);
10	
11	Description
12	
13	SendResponseFromFile
14	
15	[C#] public override void SendResponseFromFile(string filename, long offset,
16	long length);
17	[C++] public: void SendResponseFromFile(String* filename,int64 offset,
18	int64 length);
19	[VB] Overrides Public Sub SendResponseFromFile(ByVal filename As String,
20	ByVal offset As Long, ByVal length As Long)
21	[JScript] public override function SendResponseFromFile(filename : String, offset
22	: long, length : long);
23	
24	Description
25	

SendRes	oonseFromN	<b>Jemory</b>
Contractor	JOHN TOWAR	

[C#] public override void SendResponseFromMemory(byte[] data, int length); [C++] public: void SendResponseFromMemory(unsigned char data \_\_gc[], int length);

[VB] Overrides Public Sub SendResponseFromMemory(ByVal data() As Byte, ByVal length As Integer)

[JScript] public override function SendResponseFromMemory(data : Byte[], length : int);

Description

SendStatus

[C#] public override void SendStatus(int statusCode, string statusDescription);
[C++] public: void SendStatus(int statusCode, String\* statusDescription);
[VB] Overrides Public Sub SendStatus(ByVal statusCode As Integer, ByVal statusDescription As String)
[JScript] public override function SendStatus(statusCode: int, statusDescription: String);

# System.Web.Mail

Description

3

4

5

6

7

8

9

14

15

16

17

18

19

20

21

22

23

24

25

The System. Web. Mail namespace contains classes that enable you to construct and send an email attachment using the SMTP mail service built into Microsoft Windows 2000.

MailAttachment class (System.Web.Mail)

## Description

Provides properties and methods to construct an email attachment.

Constructors:

MailAttachment

Example Syntax:

[C#] public MailAttachment(string filename);

[C++] public: MailAttachment(String\* filename);

[VB] Public Sub New(ByVal filename As String)

[JScript] public function MailAttachment(filename : String); Constructs an email attachment object.

## Description

Constructs a MailAttachment object and specifies the file name of the attachment. Sets the System.Text.Encoding property to be UUEncode by default. Name of the attached file.

MailAttachment

Example Syntax:

1	
2	[C#] public MailAttachment(string filename, MailEncoding encoding);
3	[C++] public: MailAttachment(String* filename, MailEncoding encoding);
4	[VB] Public Sub New(ByVal filename As String, ByVal encoding As
5	MailEncoding)
6	[JScript] public function MailAttachment(filename : String, encoding :
7	MailEncoding);
8	
9	Description
10	Constructs an email attachment object and specifies the file name and
11	encoding of the attachment. Name of the attached file. The type of
12	System.Web.Mail.MailEncoding used by the attachment.
13	Properties:
14	Encoding
15	
16	[C#] public MailEncoding Encoding {get;}
17	[C++] public:property MailEncoding get_Encoding();
18	[VB] Public ReadOnly Property Encoding As MailEncoding
19	[JScript] public function get Encoding(): MailEncoding;
20	
21	Description
22	Indicates the type of encoding used to encode the email attachment.
23	Filename
24	
25	[C#] public string Filename {get;}

```
[C++] public: property String* get Filename();
    [VB] Public ReadOnly Property Filename As String
    [JScript] public function get Filename(): String;
    Description
           Indicates the name of the file to attach to the email.
           Methods:
           MailEncoding enumeration (System.Web.Mail)
           ToString
10
11
    Description
12
           Provides enumered values for email encoding.
13
           ToString
14
15
    [C#] public const MailEncoding Base64;
    [C++] public: const MailEncoding Base64;
17
    [VB] Public Const Base64 As MailEncoding
18
    [JScript] public var Base64 : MailEncoding;
19
20
    Description
21
           Specifies that the email message uses Base64 encoding.
22
           ToString
23
24
    [C#] public const MailEncoding UUEncode;
```

1	[C++] public: const MailEncoding UUEncode;
2	[VB] Public Const UUEncode As MailEncoding
3	[JScript] public var UUEncode : MailEncoding;
4	
5	Description
6	Specifies that the email message uses UUEncode encoding.
7	MailFormat enumeration (System.Web.Mail)
8	ToString
9	
10	
11	Description
12	Provides enumerated values for email format.
13	ToString
14	
15	[C#] public const MailFormat Html;
16	[C++] public: const MailFormat Html;
17	[VB] Public Const Html As MailFormat
18	[JScript] public var Html : MailFormat;
19	
20	Description
21	Specifies that email format is HTML.
22	ToString
23	
24	[C#] public const MailFormat Text;
25	[C++] public: const MailFormat Text;

```
[JScript] public function get Attachments(): IList;
2
    Description
3
           Specifies the list of attachments that is transmitted with the message.
           Bcc
           ToString
6
7
    [C#] public string Bcc {get; set;}
8
    [C++] public: property String* get Bcc();public: property void
    set_Bcc(String*);
10
    [VB] Public Property Bcc As String
11
    [JScript] public function get Bcc(): String; public function set Bcc(String);
12
13
    Description
14
           Gets or sets a semicolon-delimited list of email addresses that receive a
15
    Blind Carbon Copy (BCC) copy of the email message.
           The primary and carbon copy System.Web.Mail.MailMessage.Cc
17
    recipients do not see addresses included in the Bcc field.
18
           Body
19
           ToString
20
21
    [C#] public string Body {get; set;}
22
    [C++] public: property String* get Body(); public: property void
23
    set Body(String*);
24
    [VB] Public Property Body As String
```

	1	[JScript] public function get Body(): String; public function set Body(String);
	2	
	3	Description
	4	Gets or sets the body of the email message.
	5	BodyEncoding
	6	ToString
	7	
	8	[C#] public Encoding BodyEncoding {get; set;}
Tool with the state that the state of the st	9	[C++] public:property Encoding* get_BodyEncoding();public:property void
	10	set_BodyEncoding(Encoding*);
	11	[VB] Public Property BodyEncoding As Encoding
	12	[JScript] public function get BodyEncoding(): Encoding; public function set
	13	BodyEncoding(Encoding);
4	14	
	15	Description
	16	Gets or sets the encoding type of the email body.
	17	BodyFormat
	18	ToString
	19	
	20	[C#] public MailFormat BodyFormat {get; set;}
	21	[C++] public:property MailFormat get_BodyFormat();public:property void
	22	set_BodyFormat(MailFormat);
	23	[VB] Public Property BodyFormat As MailFormat
	24	[JScript] public function get BodyFormat(): MailFormat; public function set
	25	BodyFormat(MailFormat);

```
Description
           Gets or sets the content type of the email body.
3
           Cc
           ToString
6
    [C#] public string Cc {get; set;}
    [C++] public: property String* get Cc(); public: property void
    set_Cc(String*);
    [VB] Public Property Cc As String
10
    [JScript] public function get Cc(): String; public function set Cc(String);
12
    Description
13
           Gets or sets a semicolon-delimited list of email addresses that receive a
14
    Carbon Copy (CC) of the email message.
15
           From
16
           ToString
17
18
    [C#] public string From {get; set;}
    [C++] public: property String* get From();public: property void
20
    set From(String*);
21
    [VB] Public Property From As String
22
    [JScript] public function get From(): String; public function set From(String);
24
    Description
```

```
Gets or sets the email address of the sender.
           Headers
2
           ToString
3
    [C#] public IDictionary Headers {get;}
    [C++] public: property IDictionary* get Headers();
    [VB] Public ReadOnly Property Headers As IDictionary
    [JScript] public function get Headers(): IDictionary;
9
    Description
10
           Specifies the custom headers that are transmitted with the email message.
11
           Priority
12
           ToString
13
14
    [C#] public MailPriority Priority {get; set;}
15
    [C++] public: property MailPriority get Priority();public: property void
16
    set Priority(MailPriority);
17
    [VB] Public Property Priority As MailPriority
18
    [JScript] public function get Priority(): MailPriority; public function set
19
    Priority(MailPriority);
20
21
    Description
22
           Gets or sets the priority of the email message.
23
           Subject
24
           ToString
25
```

```
[C#] public string Subject {get; set;}
    [C++] public: property String* get Subject(); public: property void
    set Subject(String*);
    [VB] Public Property Subject As String
    [JScript] public function get Subject(): String; public function set Subject(String);
7
    Description
           Gets or sets the subject line of the email message.
           To
           ToString
11
12
    [C#] public string To {get; set;}
13
    [C++] public: property String* get To(); public: property void
    set_To(String*);
15
    [VB] Public Property To As String
16
    [JScript] public function get To(): String; public function set To(String);
17
18
    Description
19
           Gets or sets the email address of the recipient.
20
           UrlContentBase
21
           ToString
22
23
    [C#] public string UrlContentBase {get; set;}
    [C++] public: __property String* get_UrlContentBase();public: __property void
```

1	set_UrlContentBase(String*);
2	[VB] Public Property UrlContentBase As String
3	[JScript] public function get UrlContentBase() : String;public function set
4	UrlContentBase(String);
5	
6	Description
7	Gets or sets the URL base of all relative URLs used within the HTML
8	encoded body.
9	UrlContentLocation
10	ToString
11	
12	[C#] public string UrlContentLocation {get; set;}
13	[C++] public:property String* get_UrlContentLocation();public:property
14	<pre>void set_UrlContentLocation(String*);</pre>
15	[VB] Public Property UrlContentLocation As String
16	[JScript] public function get UrlContentLocation(): String; public function set
17	UrlContentLocation(String);
18	
19	Description
20	
21	MailPriority enumeration (System.Web.Mail)
22	ToString
23	
24	
25	Description

1	Provides enumerated values for email priority.
2	ToString
3	
4	[C#] public const MailPriority High;
5	[C++] public: const MailPriority High;
6	[VB] Public Const High As MailPriority
7	[JScript] public var High : MailPriority;
8	
9	Description
10	Specifies that the email message has high priority
11	ToString
12	
13	[C#] public const MailPriority Low;
14	[C++] public: const MailPriority Low;
15	[VB] Public Const Low As MailPriority
16	[JScript] public var Low: MailPriority;
17	
18	Description
19	Specifies that the email message has low priority.
20	ToString
21	
22	[C#] public const MailPriority Normal;
23	[C++] public: const MailPriority Normal;
24	[VB] Public Const Normal As MailPriority
25	[JScript] public var Normal : MailPriority;

11	
1	
2	Description
3	Specifies that the email message has normal priority.
4	SmtpMail class (System.Web.Mail)
5	ToString
6	
7	
8	Description
9	Provides properties and methods to send an email attachment using the
10	SMTP mail service built into Microsoft Windows 2000.
11	Mail is by default queued on a Windows 2000 system, ensuring that the
12	calling program does not block network traffic.
13	SmtpMail
14	Example Syntax:
15	ToString
16	
17	[C#] public SmtpMail();
18	[C++] public: SmtpMail();
19	[VB] Public Sub New()
20	[JScript] public function SmtpMail();
21	SmtpServer
22	ToString
23	
24	[C#] public static string SmtpServer {get; set;}
25	[C++] public:property static String* get_SmtpServer();public:property static

void set SmtpServer(String\*); [VB] Public Shared Property SmtpServer As String [JScript] public static function get SmtpServer(): String; public static function set SmtpServer(String); 5 Description Gets or sets the name of the SMTP mail server to use to send email 7 messages. 8 Send 9 10 [C#] public static void Send(MailMessage message); [C++] public: static void Send(MailMessage\* message); [VB] Public Shared Sub Send(ByVal message As MailMessage) 13 [JScript] public static function Send(message: MailMessage); 15 Description 16 Sends a mail message using arguments supplied in the MailMessage 17 properties. The System.Web.Mail.MailMessage object to send. 18 Send 19 20 [C#] public static void Send(string from, string to, string subject, string 21 messageText); 22 [C++] public: static void Send(String\* from, String\* to, String\* subject, String\* 23 messageText); 24 [VB] Public Shared Sub Send(ByVal from As String, ByVal to As String, ByVal

subject As String, ByVal messageText As String)

[JScript] public static function Send(from : String, to : String, subject : String,

messageText : String); Sends a mail message.

## Description

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Sends a mail message using the supplied destination arguments. The address of the email's sender. The address of the email's recipient. The subject line of the email message.

## **System.Web.Security**

Description

This namespace contains classes that are used to implement ASP.NET security in Web server applications.

DefaultAuthenticationEventArgs class (System.Web.Security)

## Description

Provides a wrapper around the default authentication services. This class cannot be inherited.

For more information about handling events, see .

Constructors:

DefaultAuthenticationEventArgs

Example Syntax:

1	
2	[C#] public DefaultAuthenticationEventArgs(HttpContext context);
3	[C++] public: DefaultAuthenticationEventArgs(HttpContext* context);
4	[VB] Public Sub New(ByVal context As HttpContext)
5	[JScript] public function DefaultAuthenticationEventArgs(context : HttpContext);
6	
7	Description
8	Initializes a new instance of the
9	System.Web.Security.DefaultAuthenticationEventArgs class. The context for
10	the event.
11	Properties:
12	Context
13	
14	[C#] public HttpContext Context {get;}
15	[C++] public:property HttpContext* get_Context();
16	[VB] Public ReadOnly Property Context As HttpContext
17	[JScript] public function get Context(): HttpContext;
18	
19	Description
20	The HttpContext intrinsic provides access to Request, Response, and
21	User objects.
22	Methods:
23	DefaultAuthenticationEventHandler delegate (System.Web.Security)
24	ToString
25	

Description

4

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Represents the method that handles the

DefaultAuthentication OnAuthenticate event of a

DefaultAuthenticationModule. The object that raised the event. A

DefaultAuthenticationEventArgs object that contains the event data.

When you create a

System.Web.Security.DefaultAuthenticationEventHandler delegate, you identify the method to handle the event. To associate the event with your EventHandler, add an instance of the delegate to the event. The EventHandler is called whenever the event occurs, unless you remove the delegate. For more information about EventHandler delegates, see .

DefaultAuthenticationModule class (System.Web.Security)

**ToString** 

Description

Insures that an **Authentication** object is present in the context. This class cannot be inherited.

DefaultAuthenticationModule

Example Syntax:

**ToString** 

[C#] public DefaultAuthenticationModule();

Description

```
[C++] public: DefaultAuthenticationModule();
    [VB] Public Sub New()
    [JScript] public function DefaultAuthenticationModule();
3
           ToString
4
5
    [C#] public event DefaultAuthenticationEventHandler Authenticate;
    [C++] public: event DefaultAuthenticationEventHandler* Authenticate;
    [VB] Public Event Authenticate As DefaultAuthenticationEventHandler
8
9
    Description
10
           Defines the event raised by the DefaultAuthentication module. Used by
11
    the DefaultAuthentication OnAuthenticate handler, if one exists.
12
           When you create a System. Web. Security. Default Authentication Module
13
    delegate, you identify the method to handle the event. To associate the event with
14
    your EventHandler, add an instance of the delegate to the event. The EventHandler
15
    is called whenever the event occurs, unless you remove the delegate. For more
16
    information about EventHandler delegates, see .
17
           Dispose
18
19
    [C#] public void Dispose();
20
    [C++] public: sealed void Dispose();
21
    [VB] NotOverridable Public Sub Dispose()
22
    [JScript] public function Dispose();
23
24
```

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Called by the HTTP runtime to Dispose of the module (derived from IHttpModule).

Call System.Web.Security.DefaultAuthenticationModule.Dispose when you are finished using the System.Web.Security.DefaultAuthenticationModule . The System.Web.Security.DefaultAuthenticationModule.Dispose method leaves the System.Web.Security.DefaultAuthenticationModule in an unusable state. After calling

System.Web.Security.DefaultAuthenticationModule.Dispose, you must release all references to the System.Web.Security.DefaultAuthenticationModule so the memory occupied can be reclaimed by garbage collection.

Init

```
[C#] public void Init(HttpApplication app);
```

[C++] public: \_\_sealed void Init(HttpApplication\* app);

[VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)

[JScript] public function Init(app: HttpApplication);

#### Description

Initializes the module. The HTTP application.

FileAuthorizationModule class (System.Web.Security)

**ToString** 

Description

lee@haves pilc 509-324-9256

3

5

7

8

9

10

11

12

13

15

16

17

18

20

21

22

23

24

25

Verifies that the remote user has NT permissions to access the file requested. This class cannot be inherited.

This module provides authorization services against file system ACLs. When the Windows authentication module is being used for the application, this module ensures (if in the pipeline) that the requesting user is allowed read access to the resource before executing it.

FileAuthorizationModule

Example Syntax:

**ToString** 

[C#] public FileAuthorizationModule();

[C++] public: FileAuthorizationModule();

[VB] Public Sub New()

[JScript] public function FileAuthorizationModule();

Dispose

[C#] public void Dispose();

[C++] public: sealed void Dispose();

[VB] NotOverridable Public Sub Dispose()

[JScript] public function Dispose();

Description

Disposes of the module dervied from  ${\bf IHttpModule}$  when called by the  ${\bf HttpRuntime}$  .

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Call System.Web.Security.FileAuthorizationModule.Dispose when you are finished using the System.Web.Security.FileAuthorizationModule. The System.Web.Security.FileAuthorizationModule.Dispose method leaves the System.Web.Security.FileAuthorizationModule in an unusable state. After calling System.Web.Security.FileAuthorizationModule.Dispose, you must release all references to the System.Web.Security.FileAuthorizationModule so the memory occupied can be reclaimed by garbage collection.

Init

[C#] public void Init(HttpApplication app);

[C++] public: sealed void Init(HttpApplication\* app);

[VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)

[JScript] public function Init(app: HttpApplication);

### Description

Initializes the module. The **HttpApplication** module.

FormsAuthentication class (System.Web.Security)

**ToString** 

#### Description

Provides static methods that supply helper utilities for manipulating authentication tickets. This class cannot be inherited.

**FormsAuthentication** 

Example Syntax:

1	ToString
2	
3	[C#] public FormsAuthentication();
4	[C++] public: FormsAuthentication();
5	[VB] Public Sub New()
6	[JScript] public function FormsAuthentication();
7	FormsCookieName
8	ToString
9	
10	[C#] public static string FormsCookieName {get;}
11	[C++] public:property static String* get_FormsCookieName();
12	[VB] Public Shared ReadOnly Property FormsCookieName As String
13	[JScript] public static function get FormsCookieName(): String;
14	
15	Description
16	Returns the configured cookie name used for the current application.
17	Returns a <b>String</b> .
18	FormsCookiePath
19	ToString
20	
21	[C#] public static string FormsCookiePath {get;}
22	[C++] public:property static String* get_FormsCookiePath();
23	[VB] Public Shared ReadOnly Property FormsCookiePath As String
24	[JScript] public static function get FormsCookiePath(): String;
25	

$\Gamma$	•	, .	
Desc	rın	tini	1
DUSU	ıν	$\iota\iota \iota \cup \iota$	ŧ

3

5

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Returns the configured cookie path used for the current application.

Authenticate

[C#] public static bool Authenticate(string name, string password);

[C++] public: static bool Authenticate(String\* name, String\* password);

[VB] Public Shared Function Authenticate(ByVal name As String, ByVal

password As String) As Boolean

[JScript] public static function Authenticate(name : String, password : String) :

Boolean;

### Description

Attempts to validate the credentials against those contained in the configured credential store, given the supplied credentials.

Return Value: Returns true if the credentials are valid, otherwise returns false.

The user name The user password.

Decrypt

[C#] public static FormsAuthenticationTicket Decrypt(string strEncrypted);

[C++] public: static FormsAuthenticationTicket\* Decrypt(String\* strEncrypted);

[VB] Public Shared Function Decrypt(ByVal strEncrypted As String) As

FormsAuthenticationTicket

[JScript] public static function Decrypt(strEncrypted : String) :

FormsAuthenticationTicket;

$\mathbf{r}$			, •	
De	$C \cap I$	าาท	111	n
$\mathcal{L}_{\mathcal{L}_{i}}$	$\mathcal{I} \cup \mathcal{I}$	$\iota \nu$	uc	, i i

3

5

6

7

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Returns an instance of a FormsAuthenticationTicket class, given an encrypted authentication ticket obtained from an HTTP cookie.

Return Value: Returns a FormsAuthenticationTicket object.

Encrypt

[C#] public static string Encrypt(FormsAuthenticationTicket ticket);

[C++] public: static String\* Encrypt(FormsAuthenticationTicket\* ticket);

[VB] Public Shared Function Encrypt(ByVal ticket As

FormsAuthenticationTicket) As String

[JScript] public static function Encrypt(ticket: FormsAuthenticationTicket):

String;

## Description

Produces a string containing an encrypted authentication ticket suitable for use in an HTTP cookie, given a FormsAuthenticationTicket.

Return Value: Returns a **String** containing an encrypted authentication ticket. An authentication ticket class.

#### GetAuthCookie

[C#] public static HttpCookie GetAuthCookie(string userName, bool createPersistentCookie);

[C++] public: static HttpCookie\* GetAuthCookie(String\* userName, bool createPersistentCookie);

[VB] Public Shared Function GetAuthCookie(ByVal userName As String, ByVal createPersistentCookie As Boolean) As HttpCookie

[JScript] public static function GetAuthCookie(userName : String, createPersistentCookie : Boolean) : HttpCookie; Creates an authentication cookie for a given user name.

### Description

3

5

6

8

10

11

12

13

14

15

16

17

18

20

21

22

23

Creates an authentication cookie for a given user name. This does not set the cookie as part of the outgoing response, so that an application can have more control over how the cookie is issued.

Return Value: Returns an HttpCookie. Name of the authenticated user. This does not have to map to a Windows account. Specifies whether or not a durable cookie (a cookie that is saved across browser sessions) should be issued. Cookie path defaults to'/'.

#### GetAuthCookie

[C#] public static HttpCookie GetAuthCookie(string userName, bool createPersistentCookie, string strCookiePath);

[C++] public: static HttpCookie\* GetAuthCookie(String\* userName, bool createPersistentCookie, String\* strCookiePath);

[VB] Public Shared Function GetAuthCookie(ByVal userName As String, ByVal createPersistentCookie As Boolean, ByVal strCookiePath As String) As HttpCookie

[JScript] public static function GetAuthCookie(userName : String,

create Persistent Cookie: Boolean, str Cookie Path: String): Http Cookie;

### Description

3

5

10

11

12

13

14

15

16

17

18

19

20

22

23

24

25

Creates an authentication cookie for a given user name. This does not set the cookie as part of the outgoing response, so that an application can have more control over how the cookie is issued.

Return Value: Returns an **HttpCookie**. Name of the authenticated user. This does not have to map to a Windows account. Specifies whether or not a durable cookie (a cookie that is saved across browser sessions) should be issued. Specifies cookie path.

#### GetRedirectUrl

[C#] public static string GetRedirectUrl(string userName, bool createPersistentCookie);

[C++] public: static String\* GetRedirectUrl(String\* userName, bool createPersistentCookie);

[VB] Public Shared Function GetRedirectUrl(ByVal userName As String, ByVal createPersistentCookie As Boolean) As String

[JScript] public static function GetRedirectUrl(userName : String, createPersistentCookie : Boolean) : String;

#### Description

Returns the redirect URL for the original request that caused the redirect to the logon page.

Return Value: Returns a String.

If there is no original URL, Default.aspx is used. If the redirect URL specifies a different server, this method also returns the authentication ticket as part of the query string. This method can be used by applications that need to do the redirect themselves instead of using **RedirectFromLoginPage**. Name of the authenticated user. This does not have to map to a Windows account. Specifies whether or not a durable cookie (a cookie that is saved across browser sessions) should be issued. Cookie path defaults to'/'.

HashPasswordForStoringInConfigFile

[C#] public static string HashPasswordForStoringInConfigFile(string password, string passwordFormat);

[C++] public: static String\* HashPasswordForStoringInConfigFile(String\* password, String\* passwordFormat);

[VB] Public Shared Function HashPasswordForStoringInConfigFile(ByVal password As String, ByVal passwordFormat As String) As String

[JScript] public static function HashPasswordForStoringInConfigFile(password: String, passwordFormat: String): String;

Description

Given a password and a string identifying the hash type, this routine produces a hash password suitable for storing in a configuration file.

Return Value: Returns a **String** containing a hashed password.

Password algorithms supported are SHA1 and MD5. The pasword to hash. The hash algorithm to use. Choices are "sha1" or "md5".

Initialize

1	
2	[C#] public static void Initialize();
3	[C++] public: static void Initialize();
4	[VB] Public Shared Sub Initialize()
5	[JScript] public static function Initialize();
6	
7	Description
8	Initializes FormsAuthentication by reading the configuration and getting
9	the cookie values and encryption keys for the given application.
10	RedirectFromLoginPage
11	
12	[C#] public static void RedirectFromLoginPage(string userName, bool
13	createPersistentCookie);
14	[C++] public: static void RedirectFromLoginPage(String* userName, bool
15	createPersistentCookie);
16	[VB] Public Shared Sub RedirectFromLoginPage(ByVal userName As String,
17	ByVal createPersistentCookie As Boolean)
18	[JScript] public static function RedirectFromLoginPage(userName : String,
19	createPersistentCookie: Boolean); Redirects an authenticated user back to the
20	originally requested URL.
21	
22	Description
23	Redirects an authenticated user back to the originally requested URL.

or redirects to Default.aspx if the return key does not exist. It issues an

Redirects, based on the contents of the ReturnURL key in the query string,

1 |

authentication ticket and does a **SetForms** with the ticket, using the appropriately configured cookie name for the application as part of the redirect response. Name of the user for cookie authentication purposes. This does not need to map to an account name and will be used by URL Authorization. Specifies whether or not a durable cookie (one that is saved across browser sessions) should be issued.

RedirectFromLoginPage

[C#] public static void RedirectFromLoginPage(string userName, bool createPersistentCookie, string strCookiePath);

[C++] public: static void RedirectFromLoginPage(String\* userName, bool createPersistentCookie, String\* strCookiePath);

[VB] Public Shared Sub RedirectFromLoginPage(ByVal userName As String, ByVal createPersistentCookie As Boolean, ByVal strCookiePath As String)
[JScript] public static function RedirectFromLoginPage(userName : String, createPersistentCookie : Boolean, strCookiePath : String);

#### Description

Redirects an authenticated user back to the originally requested URL.

Redirects, based on the contents of the **ReturnURL** key in the query string, or redirects to Default.aspx if the return key does not exist. It issues an authentication ticket and does a **SetForms** with the ticket, using the appropriately configured cookie name for the application as part of the redirect response. Name of the user for cookie authentication purposes. This does not need to map to an account name and will be used by URL Authorization. Specifies whether or not a

durable cookie (one that is saved across browser sessions) should be issued. Specifies cookie path. 2 RenewTicketIfOld 3 4 [C#] public static FormsAuthenticationTicket RenewTicketIfOld(FormsAuthenticationTicket tOld); [C++] public: static FormsAuthenticationTicket\* RenewTicketIfOld(FormsAuthenticationTicket\* tOld); [VB] Public Shared Function RenewTicketIfOld(ByVal tOld As FormsAuthenticationTicket) As FormsAuthenticationTicket 10 [JScript] public static function RenewTicketIfOld(tOld: 11 FormsAuthenticationTicket; FormsAuthenticationTicket; 12 13 Description 14 Conditionally updates the sliding expiration on a 15 FormsAuthenticationTicket. 16 Return Value: Returns the updated FormsAuthenticationTicket. 17 The ticket is updated with the **IssueDate** set to the current time and the 18 expiration based on the difference between the previous expiration time and issue time. The test used to determine whether the ticket should be updated is ( 20 DateTime.Now â€" IssueDate ) > ( Expiration â€" DateTime.Now ). 21 SetAuthCookie 22 23 [C#] public static void SetAuthCookie(string userName, bool

createPersistentCookie);

25

[C++] public: static void SetAuthCookie(String\* userName, bool createPersistentCookie);
[VB] Public Shared Sub SetAuthCookie(ByVal userName As String, ByVal createPersistentCookie As Boolean)
[JScript] public static function SetAuthCookie(userName: String, createPersistentCookie: Boolean); Creates an authentication ticket and attaches it to the cookie's collection of the outgoing response. It does not perform a redirect.

### Description

Creates an authentication ticket for the given *userName* and *createPersistentCookie* and attaches it to the cookie's collection of the outgoing response. It does not perform a redirect.

This has the effect of creating an authenticated user. Applications can use this when they intend to manually redirect, or if they do not want to redirect. For example, you can use this for an embedded logon area on a page. The name of an authenticated user. This does not have to map to a Windows account. Specifies whether or not a durable cookie (one that is saved across browser sessions) should be issued.

#### SetAuthCookie

[C#] public static void SetAuthCookie(string userName, bool createPersistentCookie, string strCookiePath);
[C++] public: static void SetAuthCookie(String\* userName, bool createPersistentCookie, String\* strCookiePath);
[VB] Public Shared Sub SetAuthCookie(ByVal userName As String, ByVal

createPersistentCookie As Boolean, ByVal strCookiePath As String)
[JScript] public static function SetAuthCookie(userName : String,
createPersistentCookie : Boolean, strCookiePath : String);

## Description

2

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

20

21

22

23

24

25

Creates an authentication ticket for the given userName,createPersistentCookie, and strCookiePath and attaches it to the cookie's collection of the outgoing response. It does not perform a redirect.

This has the effect of creating an authenticated user. Applications can use this when they intend to manually redirect, or if they do not want to redirect. For example, you can use this for an embedded logon area on a page. The name of an authenticated user. This does not have to map to a Windows account. Specifies whether or not a durable cookie (one that is saved across browser sessions) should be issued. Specifies cookie path.

SignOut

[C#] public static void SignOut();

[C++] public: static void SignOut();

[VB] Public Shared Sub SignOut()

[JScript] public static function SignOut();

## Description

Removes the authentication ticket by doing a **SetForms** with an empty value, given an authenticated user. This removes either durable or session cookies.

FormsAuthenticationEventArgs class (System.Web.Security)

**ToString** 1 2 3 Description Provides data for the FormsAuthentication OnAuthenticate event. This 5 class cannot be inherited. For more information about handling events, see . 7 FormsAuthenticationEventArgs 8 Example Syntax: 9 **ToString** 10 11 [C#] public FormsAuthenticationEventArgs(HttpContext context); 12 [C++] public: FormsAuthenticationEventArgs(HttpContext\* context); 13 [VB] Public Sub New(ByVal context As HttpContext) 14 [JScript] public function FormsAuthenticationEventArgs(context: HttpContext); 15 16 Description 17 Initializes a new instance of the 18 System.Web.Security.FormsAuthenticationEventArgs class. 19 The following table shows initial property values for an instance of 20 System. Web. Security. Forms Authentication Event Args. The context for the 21 event. 22 Context 23 **ToString** 24

```
1
    [C#] public HttpContext Context {get;}
    [C++] public: property HttpContext* get Context();
    [VB] Public ReadOnly Property Context As HttpContext
    [JScript] public function get Context(): HttpContext;
6
    Description
           Provides access to Request, Response, and User objects. This is the
8
    HttpContext intrinsic.
           See the ASP.NET HttpRuntime documentation for more information.
10
           User
11
           ToString
12
13
    [C#] public IPrincipal User {get; set;}
14
    [C++] public: property IPrincipal* get User(); public: property void
15
    set_User(IPrincipal*);
16
    [VB] Public Property User As IPrincipal
17
    [JScript] public function get User(): IPrincipal; public function set
18
    User(IPrincipal);
19
20
    Description
21
           Indicates the IPrincipal object to be associated with the request.
22
           The user object will be attached to the context. If User is non null and
23
    Context. User is null, the Forms Authentication Module will initialize
    Context.User with FormsAuthenticationEventArgs.User.
```

FormsAuthenticationEventHandler delegate (System.Web.Security)
ToString

# Description

Represents the method that will handle the FormsAuthentication event of a FormsAuthenticationModule. The source of the event. A System.Web.Security.FormsAuthenticationEventArgs that contains the event data.

When you create a

System.Web.Security.FormsAuthenticationEventHandler delegate, you identify the method that will handle the event. To associate the event with your event handler, add an instance of the delegate to the event. The event handler is called whenever the event occurs, unless you remove the delegate. For more information about event handler delegates, see This delegate defines the signature for the FormsAuthentication\_OnAuthenticate event handler.

FormsAuthenticationModule class (System.Web.Security)
ToString

### Description

Enables ASP.NET applications to use forms authentication. This class cannot be inherited.

FormsAuthenticationModule

Example Syntax:

	1	ToString
	2	
	3	[C#] public FormsAuthenticationModule();
	4	[C++] public: FormsAuthenticationModule();
	5	[VB] Public Sub New()
	6	[JScript] public function FormsAuthenticationModule();
	7	ToString
	8	
	9	[C#] public event FormsAuthenticationEventHandler Authenticate;
	10	[C++] public:event FormsAuthenticationEventHandler* Authenticate;
dan dan dan dan dan	11	[VB] Public Event Authenticate As FormsAuthenticationEventHandler
	12	
	13	Description
	14	Defines the event raised during authentication. This is a Global asax event
	15	that must be named FormsAuthentication_OnAuthenticate . You can use this
	16	event to customize cookie authentication.
	17	For more information about handling events, see .
	18	Dispose
	19	
	20	[C#] public void Dispose();
	21	[C++] public:sealed void Dispose();
	22	[VB] NotOverridable Public Sub Dispose()
	23	[JScript] public function Dispose();
	24	
	25	Description

2	HttpRuntime .
3	Call System.Web.Security.FormsAuthenticationModule.Dispose when
4	you are finished using the System. Web. Security. Forms Authentication Module.
5	The System.Web.Security.FormsAuthenticationModule.Dispose method leaves
6	the System. Web. Security. Forms Authentication Module in an unusable state.
7	After calling System. Web. Security. Forms Authentication Module. Dispose, you
8	must release all references to the
9	System.Web.Security.FormsAuthenticationModule so the memory it was
10	occupying can be reclaimed by garbage collection.
11	Init
12	
13	[C#] public void Init(HttpApplication app);
14	[C++] public:sealed void Init(HttpApplication* app);
15	[VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)
16	[JScript] public function Init(app: HttpApplication);
17	
18	Description
19	Initializes the module derived from IHttpModule when called by the
20	HttpRuntime. The HttpApplication module.
21	OnEnter
22	
23	[C#] public void OnEnter(object source, EventArgs eventArgs);
24	[C++] public: void OnEnter(Object* source, EventArgs* eventArgs);

Disposes of the module derived from IHttpModule when called by the

[VB] Public Sub OnEnter(ByVal source As Object, ByVal eventArgs As

1	EventArgs)
2	[JScript] public function OnEnter(source : Object, eventArgs : EventArgs);
3	
4	Description
5	Called by the HttpRuntime. The source of the event. A
6	System.Web.Security.FormsAuthenticationEventArgsthat contains the event
7	data.
8	OnLeave
9	
10	[C#] public void OnLeave(object source, EventArgs eventArgs);
11	[C++] public: void OnLeave(Object* source, EventArgs* eventArgs);
12	[VB] Public Sub OnLeave(ByVal source As Object, ByVal eventArgs As
13	EventArgs)
14	[JScript] public function OnLeave(source : Object, eventArgs : EventArgs);
15	
16	Description
17	Called by the <b>HttpRuntime</b> . The source of the event. A
18	System.Web.Security.FormsAuthenticationEventArgsthat contains the event
19	data.
20	FormsAuthenticationTicket class (System.Web.Security)
21	ToString
22	
23	
24	Description
25	

Provides the information represented in an authentication cookie as used by **FormsAuthenticationModule**. This class cannot be inherited.

For more information about using attributes, see .

FormsAuthenticationTicket

Example Syntax:

**ToString** 

[C#] public FormsAuthenticationTicket(string name, bool isPersistent, int timeout);

[C++] public: FormsAuthenticationTicket(String\* name, bool isPersistent, int timeout);

[VB] Public Sub New(ByVal name As String, ByVal isPersistent As Boolean, ByVal timeout As Integer)

[JScript] public function FormsAuthenticationTicket(name: String, isPersistent: Boolean, timeout: int);

# Description

Creates a **FormsAuthenticationTicket** instance with the specified name and cookie durability, and default values for the other settings (version set internally, dates set to current date, and expiration determined on durability of cookie).

The default settings include version number, dates that are set to the current date, and expiration as determined by the durability setting. User name associated with the ticket. If true, use a persistent cookie. Specifies the time in minutes (as an integer) for which the authentication ticket is valid.

FormsAuthenticationTicket

Example Syntax:

**ToString** 

1

2

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

[C#] public FormsAuthenticationTicket(int version, string name, DateTime issueDate, DateTime expiration, bool isPersistent, string userData);

[C++] public: FormsAuthenticationTicket(int version, String\* name, DateTime issueDate, DateTime expiration, bool isPersistent, String\* userData);

[VB] Public Sub New(ByVal version As Integer, ByVal name As String, ByVal issueDate As DateTime, ByVal expiration As DateTime, ByVal isPersistent As Boolean, ByVal userData As String)

[JScript] public function FormsAuthenticationTicket(version: int, name: String, issueDate: DateTime, expiration: DateTime, isPersistent: Boolean, userData: String); Creates a FormsAuthenticationTicket instance.

### Description

Creates a **FormsAuthenticationTicket** instance with explicit values. The version number. User name associated with the ticket. Time at which the cookie was issued. Set an expiration date for the cookie. **True** if the cookie is persistent. User-defined data to be stored in the cookie.

**FormsAuthenticationTicket** 

Example Syntax:

**ToString** 

[C#] public Forms Authentication Ticket (int version, string name, Date Time

1	issueDate, DateTime expiration, bool isPersistent, string userData, string
2	cookiePath);
3	[C++] public: FormsAuthenticationTicket(int version, String* name, DateTime
4	issueDate, DateTime expiration, bool isPersistent, String* userData, String*
5	cookiePath);
6	[VB] Public Sub New(ByVal version As Integer, ByVal name As String, ByVal
7	issueDate As DateTime, ByVal expiration As DateTime, ByVal isPersistent As
8	Boolean, ByVal userData As String, ByVal cookiePath As String)
9	[JScript] public function FormsAuthenticationTicket(version: int, name: String,
10	issueDate: DateTime, expiration: DateTime, isPersistent: Boolean, userData:
11	String, cookiePath : String);
12	
13	Description
14	Creates a FormsAuthenticationTicket instance with explicit values. The
15	version number. User name associated with the ticket. Time at which the cookie
16	was issued. Set an expiration date for the cookie. True if the cookie is persistent.
17	User-defined data to be stored in the cookie. THe path for the cookie.
18	CookiePath
19	ToString
20	
21	[C#] public string CookiePath {get;}
22	[C++] public:property String* get_CookiePath();
23	[VB] Public ReadOnly Property CookiePath As String

[JScript] public function get CookiePath(): String;

1	
2	Description
3	Returns the Path for which the cookie was issued.
4	This is used when the cookie is refreshed.
5	Expiration
6	ToString
7	
8	[C#] public DateTime Expiration {get;}
9	[C++] public:property DateTime get_Expiration();
10	[VB] Public ReadOnly Property Expiration As DateTime
11	[JScript] public function get Expiration(): DateTime;
12	
13	Description
14	Returns the date/time at which the cookie expires.
15	For durable cookies, this should be the maximum value. For session scoped
16	cookies, this should be set to the time the cookie was issued. This can be used by
17	custom application logic to implement more-advanced expiration semantics.
18	Expired
19	ToString
20	
21	[C#] public bool Expired {get;}
22	[C++] public:property bool get_Expired();
23	[VB] Public ReadOnly Property Expired As Boolean
24	[JScript] public function get Expired(): Boolean;
25	

11	
1	
2	Description
3	Returns True if the cookie has expired.
4	IsPersistent
5	ToString
6	
7	[C#] public bool IsPersistent {get;}
8	[C++] public:property bool get_IsPersistent();
9	[VB] Public ReadOnly Property IsPersistent As Boolean
10	[JScript] public function get IsPersistent(): Boolean;
11	
12	Description
13	Returns True if a durable cookie was issued. Otherwise, the authentication
14	cookie is scoped to the browser lifetime.
15	IssueDate
16	ToString
17	
18	[C#] public DateTime IssueDate {get;}
19	[C++] public:property DateTime get_IssueDate();
20	[VB] Public ReadOnly Property IssueDate As DateTime
21	[JScript] public function get IssueDate(): DateTime;
22	
23	Description
24	Returns the date/time at which the cookie was originally issued. This can be
25	used for custom expiration schemes.
	2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

Name **ToString** 2 3 [C#] public string Name {get;} [C++] public: property String\* get Name(); [VB] Public ReadOnly Property Name As String [JScript] public function get Name(): String; 8 Description 9 Returns the user name associated with the authentication cookie. 10 This string is used for authorization purposes. 11 UserData 12 **ToString** 13 14 [C#] public string UserData {get;} 15 [C++] public: property String\* get UserData(); 16 [VB] Public ReadOnly Property UserData As String 17 [JScript] public function get UserData(): String; 18 19 Description 20 Returns an application-defined string that might have been stored in the 21 cookie. 22 This field will be empty ("") if no application-defined data was provided. 23 Version 24

387

**ToString** 

25

```
1
    [C#] public int Version {get;}
    [C++] public: property int get Version();
3 .
    [VB] Public ReadOnly Property Version As Integer
    [JScript] public function get Version(): int;
5
6
    Description
7
           Returns a byte version number for future use.
8
           The current version identifier is 1.
9
           FormsIdentity class (System.Web.Security)
10
           ToString
11
12
13
    Description
14
           Provides an IIdentity- derived class to be used by
15
    FormsAuthenticationModule. It provides a way for an application to access the
16
    cookie authentication ticket. This class cannot be inherited.
17
           FormsIdentity
18
           Example Syntax:
19
           ToString
20
21
    [C#] public FormsIdentity(FormsAuthenticationTicket ticket);
22
    [C++] public: FormsIdentity(FormsAuthenticationTicket* ticket);
23
    [VB] Public Sub New(ByVal ticket As FormsAuthenticationTicket)
24
    [JScript] public function FormsIdentity(ticket: FormsAuthenticationTicket);
```

	{{	
	1	
	2	Description
	3	Initializes a new instance of the System. Web. Security. Forms I dentity
	4	class. The authentication ticket upon which this identity is based.
	5	AuthenticationType
	6	ToString
	7	
	8	[C#] public string AuthenticationType {get;}
	9	[C++] public:property String* get_AuthenticationType();
***	10	[VB] Public ReadOnly Property AuthenticationType As String
45 34 34 34 35 35 35 35 35 35 35 35 35 35 35 35 35	11	[JScript] public function get AuthenticationType(): String;
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	12	
# 15 mm	13	Description
:: :: *: :: a;	14	The type of the identity (in this case, "Forms").
10 mm	15	IsAuthenticated
	16	ToString
4	17	
	18	[C#] public bool IsAuthenticated {get;}
	19	[C++] public:property bool get_IsAuthenticated();
	20	[VB] Public ReadOnly Property IsAuthenticated As Boolean
	21	[JScript] public function get IsAuthenticated(): Boolean;
	22	
	23	Description
	24	Indicates whether authentication took place.

Name

1	ToString
2	
3	[C#] public string Name {get;}
4	[C++] public:property String* get_Name();
5	[VB] Public ReadOnly Property Name As String
6	[JScript] public function get Name() : String;
7	
8	Description
9	The name of the identity (in this case, the user name).
10	Ticket
11	ToString
12	
13	[C#] public FormsAuthenticationTicket Ticket {get;}
14	[C++] public:property FormsAuthenticationTicket* get_Ticket();
15	[VB] Public ReadOnly Property Ticket As FormsAuthenticationTicket
16	[JScript] public function get Ticket(): FormsAuthenticationTicket;
17	
18	Description
19	Returns the FormsAuthenticationTicket associated with the current
20	request.
21	PassportAuthenticationEventArgs class (System.Web.Security)
22	ToString
23	
24	
25	Description

1 The event argument passed to the PassportAuthentication OnAuthenticate event by the PassportAuthentication module. Since there is already an identity at this point, this is useful mainly for 3 attaching a custom **IPrincipal** object to the context using the supplied identity. For more information about handling events, see. 5 PassportAuthenticationEventArgs 6 Example Syntax: **ToString** 8 9 The course of th [C#] public PassportAuthenticationEventArgs(PassportIdentity identity, 10 HttpContext context); 11 [C++] public: PassportAuthenticationEventArgs(PassportIdentity\* identity, HttpContext\* context); 13 [VB] Public Sub New(ByVal identity As PassportIdentity, ByVal context As 14 HttpContext) 15 [JScript] public function PassportAuthenticationEventArgs(identity: 16 PassportIdentity, context: HttpContext); 17 18 Description 19 Initializes a new instance of the 20 System. Web. Security. Passport Authentication Event Args class. The identity 21 object The context for the event. 22 Context 23 **ToString** 24

```
[C#] public HttpContext Context {get;}
    [C++] public: property HttpContext* get Context();
    [VB] Public ReadOnly Property Context As HttpContext
    [JScript] public function get Context(): HttpContext;
6
    Description
           The HttpContext intrinsic provides access to Request, Response, and
8
    User objects.
           Identity
10
           ToString
11
12
    [C#] public PassportIdentity Identity {get;}
13
    [C++] public: property PassportIdentity* get_Identity();
    [VB] Public ReadOnly Property Identity As PassportIdentity
15
    [JScript] public function get Identity(): PassportIdentity;
16
17
    Description
18
           An authenticated Passport identity.
19
           User
20
           ToString
21
22
    [C#] public IPrincipal User {get; set;}
23
    [C++] public: __property IPrincipal* get_User();public: __property void
    set User(IPrincipal*);
25
```

[VB] Public Property User As IPrincipal 1 [JScript] public function get User(): IPrincipal; public function set 2 User(IPrincipal); 3 Description 5 Associates an **IPrincipal** object with the request. 6 The User object should be attached to the context. If User is non-null and 7 Context.User is null, the PassportAuthenticationModule will initialize 8 Context. User with PassportAuthentication EventArgs. User. 9 PassportAuthenticationEventHandler delegate (System.Web.Security) 10 **ToString** 11 12 13 Description 14 Represents the method that handles the 15 PassportAuthentication\_OnAuthenticate event of a 16 **PassportAuthenticationModule**. The object that raised the event. A 17 **PassportAuthenticationEventArgs** object that contains the event data. 18 When you create a 19 System. Web. Security. Passport Authentication Event Handler delegate, you 20 identify the method to handle the event. To associate the event with your 21 EventHandler, add an instance of the delegate to the event. The EventHandler is 22 called whenever the event occurs, unless you remove the delegate. For more 23 information about EventHandler delegates, see . 24

lee®hayes piic 509+324+9255 393

PassportAuthenticationModule class (System.Web.Security)

ToString

Description

1

2

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Provides a wrapper around PassportAuthentication services. This class cannot be inherited.

Installation of the Passport SDK is still required, as is registration of the site with the Passport authority. The **PassportAuthentication\_OnAuthenticate** event is raised for applications that are designed to attach a custom **IPrincipal** object to the context. The Passport service itself does the authentication, so that cannot be overridden.

PassportAuthenticationModule

Example Syntax:

**ToString** 

[C#] public PassportAuthenticationModule();

[C++] public: PassportAuthenticationModule();

[VB] Public Sub New()

[JScript] public function PassportAuthenticationModule();

**ToString** 

[C#] public event PassportAuthenticationEventHandler Authenticate;

 $[C++]\ public: \underline{\quad} event\ PassportAuthenticationEventHandler*\ Authenticate;$ 

[VB] Public Event Authenticate As PassportAuthenticationEventHandler

Description

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Raised during authentication. This is a Global.asax event that must be named PassportAuthentication OnAuthenticate event.

Advanced users use this event to attach a custom **IPrinciple** object to the context.

Dispose

[C#] public void Dispose();

[C++] public: \_\_sealed void Dispose();

[VB] NotOverridable Public Sub Dispose()

[JScript] public function Dispose();

Description

Disposes of the module derived from **IHttpModule** when called by the **HttpRuntime**.

Call System.Web.Security.PassportAuthenticationModule.Dispose when you are finished using the

System. Web. Security. Passport Authentication Module. The

 ${\bf System. Web. Security. Passport Authentication Module. Dispose \ method\ leaves}$ 

the System. Web. Security. Passport Authentication Module in an unusable state.

 $After\ calling\ System. Web. Security. Passport Authentication Module. Dispose\ ,$ 

you must release all references to the

System. Web. Security. Passport Authentication Module so the memory it occupied can be reclaimed by garbage collection.

1	Init
2	
3	[C#] public void Init(HttpApplication app);
4	[C++] public:sealed void Init(HttpApplication* app);
5	[VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)
6	[JScript] public function Init(app: HttpApplication);
7	
8	Description
9	Initializes the module (derived from IHttpModule) when called by the
10	HttpRuntime . The HttpApplication module
11	PassportIdentity class (System.Web.Security)
12	ToString
13	
14	
15	Description
16	Provides access to the Passport profile information contained in the
17	Passport profile cookies. This is an <b>IIdentity-</b> derived class. This class cannot be
18	inherited.
19	ASP.NET Beta 2 requires the 1.4 version of the Passport SDK.
20	PassportIdentity
21	Example Syntax:
22	ToString
23	
24	[C#] public PassportIdentity();
25	[C++] public: PassportIdentity();

[JScript] public function get Error(): int;

24

25

The state of the s
7
t af
: 21. :21.
*
¥::
#: #:
<b>.</b>

7	٠	4: -	
Descr	ъ	uo	n

3

5

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Returns an error state associated with the current Passport ticket. For more information, see the error property in the Passport documentation.

GetFromNetworkServer

**ToString** 

[C#] public bool GetFromNetworkServer {get;}

[C++] public: property bool get\_GetFromNetworkServer();

[VB] Public ReadOnly Property GetFromNetworkServer As Boolean

[JScript] public function get GetFromNetworkServer(): Boolean;

# Description

Returns true if a connection is coming back from the Passport server (logon, update, or registration) and if the Passport data contained on the query string is valid.

HasSavedPassword

**ToString** 

[C#] public bool HasSavedPassword {get;}

[C++] public: property bool get HasSavedPassword();

[VB] Public ReadOnly Property HasSavedPassword As Boolean

[JScript] public function get HasSavedPassword(): Boolean;

Description

. 2
N 18 18 18
1 20
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2,5
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
( #) ( p)
1 2
::
***
11
: 4:
2
1 30 F
: #=

	1	Returns true if the Passport member's ticket indicates that the password				
	2	was saved on the Passport logon page the last time the ticket was refreshed.				
	3	For more information, see the Passport SDK documentation.				
	4	HasTicket				
	5	ToString				
	6					
	7	[C#] public bool HasTicket {get;}				
	8	[C++] public:property bool get_HasTicket();				
	9	[VB] Public ReadOnly Property HasTicket As Boolean				
**************************************	10	[JScript] public function get HasTicket(): Boolean;				
	11					
	12	Description				
the state of the s	13	Returns true if there is a Passport ticket as a cookie on the query string.				
;	14	For more information, see the Passport SDK documentation.				
A CAN COLOR OF THE CAN	15	IsAuthenticated				
	16	ToString				
#=	17					
	18	[C#] public bool IsAuthenticated {get;}				
	19	[C++] public:property bool get_IsAuthenticated();				
	20	[VB] Public ReadOnly Property IsAuthenticated As Boolean				
	21	[JScript] public function get IsAuthenticated(): Boolean;				
	22					
	23	Description				
	24	Returns true if the user is authenticated by a Passport authority.				
	25	For more information, see the Passport SDK documentation .				

The state of the s
The Hard of the Control of the Hard of the

1	Item
2	ToString
3	
4	[C#] public string this[string strProfileName] {get;}
5	[C++] public:property String* get_Item(String* strProfileName);
6	[VB] Public Default ReadOnly Property Item(ByVal strProfileName As String)
7	As String
8	[JScript] returnValue = PassportIdentityObject.Item(strProfileName);
9	
10	Description
11	Acceses Passport profile attributes . Calling this property is the equivalent
12	of calling GetProfileObject or SetProfileObject.
13	For more information, see the Passport SDK documentation. The Passport
14	profile attribute to query.
15	Name
16	ToString
17	
18	[C#] public string Name {get;}
19	[C++] public:property String* get_Name();
20	[VB] Public ReadOnly Property Name As String
21	[JScript] public function get Name(): String;
22	
23	Description
24	Consists of the name of the identity. In this case, it is the value of the
25	Passnort PLIID

	11	
	1	For more information, see the Passport SDK documentation.
	2	TicketAge
	3	ToString
	4	
	5	[C#] public int TicketAge {get;}
	6	[C++] public:property int get_TicketAge();
	7	[VB] Public ReadOnly Property TicketAge As Integer
	8	[JScript] public function get TicketAge(): int;
	9	
And the	10	Description
was and and dad	11	Consists of the time, in seconds, since the last ticket was issued or
	12	refreshed.
	13	For more information, see the Passport SDK documentation.
	14	TimeSinceSignIn
	15	ToString
	16	
	17	[C#] public int TimeSinceSignIn {get;}
	18	[C++] public:property int get_TimeSinceSignIn();
	19	[VB] Public ReadOnly Property TimeSinceSignIn As Integer
	20	[JScript] public function get TimeSinceSignIn(): int;
	21	
	22	Description
	23	Consists of the time, in seconds, since a member's log on to the Passport
	24	logon server.
	25	For more information, see the Passport SDK documentation.

		.1	r	т 1
Α	11	tn	11	111

[C#] public string AuthUrl();

[C++] public: String\* AuthUrl();

[VB] Public Function AuthUrl() As String

[JScript] public function AuthUrl(): String; There are two overloads for this method.

# Description

Returns a string containing the login server URL for a member, along with the optional information sent to the login server in the query string.

AuthUrl

[C#] public string AuthUrl(string strReturnUrl, int iTimeWindow, bool fForceLogin, string strCoBrandedArgs, int iLangID, string strNameSpace, int iKPP, bool bUseSecureAuth);

[C++] public: String\* AuthUrl(String\* strReturnUrl, int iTimeWindow, bool fForceLogin, String\* strCoBrandedArgs, int iLangID, String\* strNameSpace, int iKPP, bool bUseSecureAuth);

[VB] Public Function AuthUrl(ByVal strReturnUrl As String, ByVal iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal bUseSecureAuth As Boolean) As String [JScript] public function AuthUrl(strReturnUrl: String, iTimeWindow: int,

lee@haves pik: 509-324-9255 402 MS1-863US.APP

fForceLogin: Boolean, strCoBrandedArgs: String, iLangID: int, strNameSpace:

String, iKPP: int, bUseSecureAuth: Boolean): String;

|| || Description

Similar to AuthUrl(). Returns the authentication server URL for a member.

For more information about the parameters for this method, see the Passport SDK documentation. Sets the URL of the location that the Login server should redirect to after sign-in is complete. Specifies the interval during which members must have last signed in. Determines how the iTimeWindow will be used. Specifies variables to be appended to the URL of the Co-branding Template script page that was specified at initial participant registration. Specifies the language in which the required domain authority page should be displayed. Specifies the domain in which the Passport should be created. Specifies data collection policies for purposes of COPPA compliance. Declares whether the actual sign-in UI should be served HTTPS from the Passport domain authority.

AuthUrl

[C#] public string AuthUrl(string strReturnUrl, int iTimeWindow, int iForceLogin, string strCoBrandedArgs, int iLangID, string strNameSpace, int iKPP, int iUseSecureAuth);

[C++] public: String\* AuthUrl(String\* strReturnUrl, int iTimeWindow, int iForceLogin, String\* strCoBrandedArgs, int iLangID, String\* strNameSpace, int iKPP, int iUseSecureAuth);

[VB] Public Function AuthUrl(ByVal strReturnUrl As String, ByVal iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal strNameSpace

17

18

19

20

21

22

23

24

25

As String, ByVal iKPP As Integer, ByVal iUseSecureAuth As Integer) As String [JScript] public function AuthUrl(strReturnUrl: String, iTimeWindow: int, 2 iForceLogin: int, strCoBrandedArgs: String, iLangID: int, strNameSpace: 3 String, iKPP: int, iUseSecureAuth: int): String; 4 5 Description 6 7 AuthUrl2 8 9 [C#] public string AuthUrl2(); 10 [C++] public: String\* AuthUrl2(); 11 [VB] Public Function AuthUrl2() As String 12 [JScript] public function AuthUrl2(): String; Retrieves a String containing the 13 logon server URL for a member, as well as the optional information sent to the 14 logon server in the query string. 15

Description

Retrieves a **String** containing the logon server URL for a member, as well as the optional information sent to the logon server in the query string.

Return Value: The logon server URL for a member, as well as the optional information sent to the logon server in the query string.

This URL can be used to generate a link for a member who has not signed in previously or who has an expired ticket. For more details, see

IPassportManager2::Authaurl2 in the Passport SDK documentation.

AuthUrl2

[C#] public string AuthUrl2(string strReturnUrl, int iTimeWindow, bool fForceLogin, string strCoBrandedArgs, int iLangID, string strNameSpace, int iKPP, bool bUseSecureAuth);
[C++] public: String\* AuthUrl2(String\* strReturnUrl, int iTimeWindow, bool fForceLogin, String\* strCoBrandedArgs, int iLangID, String\* strNameSpace, int iKPP, bool bUseSecureAuth);
[VB] Public Function AuthUrl2(ByVal strReturnUrl As String, ByVal iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal bUseSecureAuth As Boolean) As String [JScript] public function AuthUrl2(strReturnUrl: String, iTimeWindow: int, fForceLogin: Boolean, strCoBrandedArgs: String, iLangID: int, strNameSpace:

#### Description

Retrieves a **String** containing the logon server URL for a member, as well as the optional information sent to the logon server in the query string.

Return Value: The logon server URL for a member, as well as the optional information sent to the logon server in the query string.

String, iKPP: int, bUseSecureAuth: Boolean): String;

This URL can be used to generate a link for a member who has not signed in previously or who has an expired ticket. For more details, see

IPassportManager2::AuthUrl2 in the Passport SDK documentation. See

Passport documentation for IPassportManager2::AuthUrl2. See Passport documentation

for IPassportManager2::AuthUrl2. See Passport documentation for IPassportManager2::AuthUrl2. See Passport documentation for IPassportManager2::AuthUrl2. See Passport documentation for IPassportManager2::AuthUrl2. See Passport documentation for IPassportManager2::AuthUrl2. See Passport documentation for IPassportManager2::AuthUrl2. See Passport documentation for IPassportManager2::AuthUrl2.

AuthUrl2

[C#] public string AuthUrl2(string strReturnUrl, int iTimeWindow, int iForceLogin, string strCoBrandedArgs, int iLangID, string strNameSpace, int iKPP, int iUseSecureAuth);

[C++] public: String\* AuthUrl2(String\* strReturnUrl, int iTimeWindow, int iForceLogin, String\* strCoBrandedArgs, int iLangID, String\* strNameSpace, int iKPP, int iUseSecureAuth);

[VB] Public Function AuthUrl2(ByVal strReturnUrl As String, ByVal iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal strNameSpace strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal iUseSecureAuth As Integer) As String [JScript] public function AuthUrl2(strReturnUrl: String, iTimeWindow: int, iForceLogin: int, strCoBrandedArgs: String, iLangID: int, strNameSpace:

String, iKPP: int, iUseSecureAuth: int): String;

Description

Retrieves a **String** containing the logon server URL for a member, as well as the optional information sent to the logon server in the query string.

Return Value: The logon server URL for a member, as well as the optional information sent to the logon server in the query string.

You can use this URL to generate a link for a member who has not previously signed in or who has an expired ticket. For more details, see IPassportManager2::AuthUrl2 in the Passport SDK documentation. See Passport documentation for IPassportManager2::AuthUrl2. See Passport documentation for IPassportManager2::AuthUrl2.

Compress

[C#] public static string Compress(string strData);

[C++] public: static String\* Compress(String\* strData);

[VB] Public Shared Function Compress(ByVal strData As String) As String

[JScript] public static function Compress(strData : String) : String;

Description

Compresses data.

Return Value: Compressed data.

For more information about **IPassportCrypt::Compress**, see the Passport SDK documentation. Data to be compressed.

CryptIsValid

1	
2	[C#] public static bool CryptIsValid();
3	[C++] public: static bool CryptIsValid();
4	[VB] Public Shared Function CryptIsValid() As Boolean
5	[JScript] public static function CryptIsValid() : Boolean;
6	
7	Description
8	Returns true if the key used for encryption and decryption is valid and if
9	the Passport Manager object is in a valid state for encryption.
10	See the Passport SDK documentation for IPassportCrypt::get_IsValid.
11	CryptPutHost
12	
13	[C#] public static int CryptPutHost(string strHost);
14	[C++] public: static int CryptPutHost(String* strHost);
15	[VB] Public Shared Function CryptPutHost(ByVal strHost As String) As Integer
16	[JScript] public static function CryptPutHost(strHost : String) : int;
17	
18	Description
19	Sets the key being used by the current Passport Crypt object by referring to
20	the host name or IP address used by the desired installation.
21	See the Passport SDK documentation for IPassportCrypt::put_Host.
22	Host name or IP address.
23	CryptPutSite
24	
25	[C#] public static int CryptPutSite(string strSite);

1	[C++] public: static int CryptPutSite(String* strSite);
2	[VB] Public Shared Function CryptPutSite(ByVal strSite As String) As Integer
3	[JScript] public static function CryptPutSite(strSite : String) : int;
4	
5	Description
6	Sets the key being used by the current Passport Crypt object by referring to
7	the site-name label assigned to that key when the key was first installed.
8	See the Passport SDK documentation for IPassportCrypt::put_Site.
9	Decompress
10	
11	[C#] public static string Decompress(string strData);
12	[C++] public: static String* Decompress(String* strData);
13	[VB] Public Shared Function Decompress(ByVal strData As String) As String
14	[JScript] public static function Decompress(strData : String) : String;
15	
16	Description
17	Decompresses data that has been compressed by the Compress method
18	Return Value: Decompressed data.
19	See the Passport SDK documentation for IPassportCrypt::Compress.
20	Data to be decompressed.
21	Decrypt
22	
23	[C#] public static string Decrypt(string strData);
24	[C++] public: static String* Decrypt(String* strData);
25	[VB] Public Shared Function Decrypt(ByVal strData As String) As String

[JScript] public static function Decrypt(strData : String) : String; 2 Description 3 Decrypts data using the Passport participant key for the current site. 4 Return Value: Decrypted data. 5 See the Passport SDK documentation for IPassportCrypt::Decrypt . The 6 data to be decrypted. 7 Encrypt 8 9 [C#] public static string Encrypt(string strData); 10 [C++] public: static String\* Encrypt(String\* strData); 11 [VB] Public Shared Function Encrypt(ByVal strData As String) As String 12 [JScript] public static function Encrypt(strData : String) : String; 13 14 Description 15 Encrypts data using the Passport participant key for the current site. 16 Maximum input size is 2045 characters. 17 Return Value: Encrypted data. 18 See the Passport SDK documentation for IPassportCrypt::Encrypt . The 19 data to be encrypted. 20 Finalize 21 22 [C#] ~PassportIdentity(); 23 [C++] ~PassportIdentity(); 24 [VB] Overrides Protected Sub Finalize()

[JScript] protected override function Finalize(); 2 Description 3 Calls Finalize method. GetDomainAttribute 5 6 [C#] public string GetDomainAttribute(string strAttribute, int iLCID, string strDomain); 8 [C++] public: String\* GetDomainAttribute(String\* strAttribute, int iLCID, String\* 9 strDomain); 10 [VB] Public Function GetDomainAttribute(ByVal strAttribute As String, ByVal 11 iLCID As Integer, ByVal strDomain As String) As String 12 [JScript] public function GetDomainAttribute(strAttribute: String, iLCID: int, 13 strDomain: String): String; 14 15 Description 16 Provides information for a Passport domain by querying the Passport CCD 17 for the requested **DomainAttribute**. 18 For more information, see the Passport SDK documentation . The name of 19 the attribute value to retrieve. Specifies the language in which various Passport 20 network pages should be displayed to the member. The domain authority name to 21 query for an attribute. 22 GetDomainFromMemberName 23 24 [C#] public string GetDomainFromMemberName(string strMemberName);

411

1	[C++] public: String* GetDomainFromMemberName(String* strMemberName);
2	[VB] Public Function GetDomainFromMemberName(ByVal strMemberName As
3	String) As String
4	[JScript] public function GetDomainFromMemberName(strMemberName:
5	String): String;
6	
7	Description
8	Returns the Passport domain from the member-name string. Name of the
9	Passport member
10	GetIsAuthenticated
11	
12	[C#] public bool GetIsAuthenticated(int iTimeWindow, bool bForceLogin, bool
13	bCheckSecure);
14	[C++] public: bool GetIsAuthenticated(int iTimeWindow, bool bForceLogin, boo
15	bCheckSecure);
16	[VB] Public Function GetIsAuthenticated(ByVal iTimeWindow As Integer,
17	ByVal bForceLogin As Boolean, ByVal bCheckSecure As Boolean) As Boolean
18	[JScript] public function GetIsAuthenticated(iTimeWindow: int, bForceLogin:
19	Boolean, bCheckSecure: Boolean): Boolean; Returns true if the user is
20	authenticated by a Passport authority.
21	
22	Description
23	Returns true if the user is authenticated by a Passport authority.
24	Return Value: true if the user is authenticated by a Passport authority.
25	

See passport documentation for **IPassportManager::IsAuthenticated**. Specifies the interval during which members must have last signed in to the calling domain. Determines how *iTimeWindow* is used. Enables checking for a secure logon.

#### GetIsAuthenticated

[C#] public bool GetIsAuthenticated(int iTimeWindow, int iForceLogin, int iCheckSecure);

[C++] public: bool GetIsAuthenticated(int iTimeWindow, int iForceLogin, int iCheckSecure);

[VB] Public Function GetIsAuthenticated(ByVal iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal iCheckSecure As Integer) As Boolean [JScript] public function GetIsAuthenticated(iTimeWindow: int, iForceLogin: int, iCheckSecure: int): Boolean;

#### Description

Returns **true** if the user is authenticated by a Passport authority.

Return Value: **true** if the user is authenticated by a Passport authority.

See Passport documentation for **IPassportManager::IsAuthenticated**. Specifies the interval during which members must have last signed in to the calling domain. Determines how *iTimeWindow* is used. Enables checking for a secure logon.

### **GetProfileObject**

[C#] public object GetProfileObject(string strProfileName);

1	[C++] public: Object* GetProfileObject(String* strProfileName);
2	[VB] Public Function GetProfileObject(ByVal strProfileName As String) As
3	Object
4	[JScript] public function GetProfileObject(strProfileName : String) : Object;
5	
6	Description
7	Returns Passport profile information for the supplied profile attribute. The
8	Passport profile attribute to query.
9	HasFlag
10	
11	[C#] public bool HasFlag(int iFlagMask);
12	[C++] public: bool HasFlag(int iFlagMask);
13	[VB] Public Function HasFlag(ByVal iFlagMask As Integer) As Boolean
14	[JScript] public function HasFlag(iFlagMask: int): Boolean;
15	
16	Description
17	Returns <b>true</b> if a given flag is set in this user's profile.
18	For more information, see the Passport SDK documentation. The Passport
19	profile flag to query.
20	HasProfile
21	
22	[C#] public bool HasProfile(string strProfile);
23	[C++] public: bool HasProfile(String* strProfile);
24	[VB] Public Function HasProfile(ByVal strProfile As String) As Boolean
25	[JScript] public function HasProfile(strProfile: String): Boolean;

Des	crin	tion
Desc	crip	uon

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Returns true if a given profile attribute exists in this user's profile.

For more information, see the Passport SDK documentation . The Passport profile attribute to query.

HaveConsent

[C#] public bool HaveConsent(bool bNeedFullConsent, bool bNeedBirthdate);

[C++] public: bool HaveConsent(bool bNeedFullConsent, bool bNeedBirthdate);

[VB] Public Function HaveConsent(ByVal bNeedFullConsent As Boolean, ByVal

bNeedBirthdate As Boolean) As Boolean

[JScript] public function HaveConsent(bNeedFullConsent : Boolean,

bNeedBirthdate: Boolean): Boolean;

## Description

Returns true if Full Consent is granted in this user's profile.

For more information, see the Passport SDK documentation . true if Full Consent is required for Passport Authentication.

LoginUser

[C#] public int LoginUser();

[C++] public: int LoginUser();

[VB] Public Function LoginUser() As Integer

[JScript] public function LoginUser(): int;

# Description Logs

Logs the user on, either by generating a 302-redirect URL or initiating a Passport-aware client authentication exchange.

This method supports Passport-aware client applications. Note that there are some policy restrictions on the use of this method in certain scenarios. Note also that this method should not be called from a frameset, as suggested in the Passport documentation. Finally, this method requires that Msppfltr.dll be properly installed and running as an ISAPI filter at the global level of the participant Web site.

LoginUser

[C#] public int LoginUser(string szRetURL, int iTimeWindow, bool fForceLogin, string szCOBrandArgs, int iLangID, string strNameSpace, int iKPP, bool fUseSecureAuth, object oExtraParams);
[C++] public: int LoginUser(String\* szRetURL, int iTimeWindow, bool fForceLogin, String\* szCOBrandArgs, int iLangID, String\* strNameSpace, int iKPP, bool fUseSecureAuth, Object\* oExtraParams);
[VB] Public Function LoginUser(ByVal szRetURL As String, ByVal iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal szCOBrandArgs As String, ByVal iLangID As Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal fUseSecureAuth As Boolean, ByVal oExtraParams As Object) As Integer

lee@hayes pilc 509-324-9256 416 MS1-863US.APP

[JScript] public function LoginUser(szRetURL : String, iTimeWindow : int,

fForceLogin: Boolean, szCOBrandArgs: String, iLangID: int, strNameSpace:

String, iKPP: int, fUseSecureAuth: Boolean, oExtraParams: Object): int; Logs the user on, either by generating a 302-redirect URL or initiating a Passport-aware client authentication exchange.

## Description

Logs the user on, either by generating a 302-redirect URL or initiating a Passport-aware client authentication exchange.

This method supports Passport-aware client applications. Note that there are some policy restrictions on the use of this method in certain scenarios. Note also that this method should not be called from a frameset, as suggested in the Passport documentation. Finally, this method requires that Msppfltr.dll be properly installed and running as an ISAPI filter at the global level of the participant Web site. See Passport documentation for IPassportManager2::LoginUser . See Passport

LoginUser

lee@hayes plic 509+324+9256

[C#] public int LoginUser(string szRetURL, int iTimeWindow, int fForceLogin, string szCOBrandArgs, int iLangID, string strNameSpace, int iKPP, int iUseSecureAuth, object oExtraParams);

[C++] public: int LoginUser(String\* szRetURL, int iTimeWindow, int

14

15

16

17

18

19

20

21

22

23

24

fForceLogin, String\* szCOBrandArgs, int iLangID, String\* strNameSpace, int 1 2 3 4 5 6 Object) As Integer 7 8 9 iKPP: int, iUseSecureAuth: int, oExtraParams: Object): int; 10 11 Description 12

iKPP, int iUseSecureAuth, Object\* oExtraParams); [VB] Public Function LoginUser(ByVal szRetURL As String, ByVal iTimeWindow As Integer, ByVal fForceLogin As Integer, ByVal szCOBrandArgs As String, ByVal iLangID As Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal iUseSecureAuth As Integer, ByVal oExtraParams As [JScript] public function LoginUser(szRetURL : String, iTimeWindow : int, fForceLogin: int, szCOBrandArgs: String, iLangID: int, strNameSpace: String,

Logs the user on, either by generating a 302-redirect URL or initiating a Passport-aware client authentication exchange.

This method supports Passport-aware client applications. Note that there are some policy restrictions on the use of this method in certain scenarios. Note also that this method should not be called from a frameset, as suggested in the Passport documentation. Finally, this method requires that Msppfltr.dll be properly installed and running as an ISAPI filter at the global level of the participant Web site. See Passport documentation for IPassportManager2::LoginUser . See Passport

25

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

documentation for IPassportManager2::LoginUser . See Passport documentation for IPassportManager2::LoginUser. LogoTag [C#] public string LogoTag(); [C++] public: String\* LogoTag(); [VB] Public Function LogoTag() As String [JScript] public function LogoTag(): String; Description Returns an HTML snippet containing an image tag for a Passport link. This is based on the current state of the identity (already signed in, and such). LogoTag [C#] public string LogoTag(string strReturnUrl, int iTimeWindow, bool fForceLogin, string strCoBrandedArgs, int iLangID, bool fSecure, string strNameSpace, int iKPP, bool bUseSecureAuth); [C++] public: String\* LogoTag(String\* strReturnUrl, int iTimeWindow, bool fForceLogin, String\* strCoBrandedArgs, int iLangID, bool fSecure, String\* strNameSpace, int iKPP, bool bUseSecureAuth); [VB] Public Function LogoTag(ByVal strReturnUrl As String, ByVal iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal

strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal fSecure As

Boolean, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal

bUseSecureAuth As Boolean) As String

[JScript] public function LogoTag(strReturnUrl: String, iTimeWindow: int, fForceLogin: Boolean, strCoBrandedArgs: String, iLangID: int, fSecure: Boolean, strNameSpace: String, iKPP: int, bUseSecureAuth: Boolean): String;

## Description

Similar to LogoTag(), this method returns an HTML snippet for the Passport Logo for the current member

For additional details on the parameters for this method, see the Passport SDK documentation. Sets the URL of the location that the Login server should redirect members to after sign-in is complete. Specifies the interval during which members must have last signed in. Determines how iTimeWindow gets used. Specifies variables to be appended as query string variables to the URL of the participant's Co-branding Template script page. Specifies the language to be used for the Login page that is displayed to the member. Declares whether this method is being called from an HTTPS (SSL) page. Specifies the domain in which the Passport should be created. Specifies data collection policies for purposes of COPPA compliance. Declares whether the actual sign-in UI should be served HTTPS from the Passport domain authority.

LogoTag

[C#] public string LogoTag(string strReturnUrl, int iTimeWindow, int iForceLogin, string strCoBrandedArgs, int iLangID, int iSecure, string strNameSpace, int iKPP, int iUseSecureAuth);
[C++] public: String\* LogoTag(String\* strReturnUrl, int iTimeWindow, int iForceLogin, String\* strCoBrandedArgs, int iLangID, int iSecure, String\*

	strNameSpace, int iKPP, int iUseSecureAuth);
1	
2	[VB] Public Function LogoTag(ByVal strReturnUrl As String, ByVal
3	iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal
4	strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal iSecure As
5	Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal
6	iUseSecureAuth As Integer) As String
7	[JScript] public function LogoTag(strReturnUrl : String, iTimeWindow : int,
8	iForceLogin: int, strCoBrandedArgs: String, iLangID: int, iSecure: int,
9	strNameSpace: String, iKPP: int, iUseSecureAuth: int): String;
10	
11	Description
12	
13	LogoTag2
14	
15	[C#] public string LogoTag2();
16	[C++] public: String* LogoTag2();
17	[VB] Public Function LogoTag2() As String
18	[JScript] public function LogoTag2(): String; Returns an HTML fragment
19	containing an < img > tag for a Passport link. This is based on the current state of
20	the identity (already logged on, and so on).
21	
22	Description
23	Returns an HTML fragment containing an < img > tag for a Passport link.
24	This is based on the current state of the identity (already logged on, and so on).

9

11

3

4

5

The link image displays either Sign In if no valid ticket cookie is detected, or Sign Out if a valid cookie ticket is detected. For more details, see IPassportManager2::LogoTag2 in the Passport SDK documentation.

LogoTag2

[C#] public string LogoTag2(string strReturnUrl, int iTimeWindow, bool fForceLogin, string strCoBrandedArgs, int iLangID, bool fSecure, string strNameSpace, int iKPP, bool bUseSecureAuth);

[C++] public: String\* LogoTag2(String\* strReturnUrl, int iTimeWindow, bool fForceLogin, String\* strCoBrandedArgs, int iLangID, bool fSecure, String\* strNameSpace, int iKPP, bool bUseSecureAuth);

[VB] Public Function LogoTag2(ByVal strReturnUrl As String, ByVal iTimeWindow As Integer, ByVal fForceLogin As Boolean, ByVal strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal fSecure As Boolean, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal bUseSecureAuth As Boolean) As String

[JScript] public function LogoTag2(strReturnUrl: String, iTimeWindow: int,

fForceLogin: Boolean, strCoBrandedArgs: String, iLangID: int, fSecure:

Boolean, strNameSpace: String, iKPP: int, bUseSecureAuth: Boolean): String;

# Description

22

23

24

25

Returns an HTML fragment containing an < img > tag for a Passport link. This is based on the current state of the identity (already logged on, and so on).

The link image displays either Sign In if no valid ticket cookie is detected, or Sign Out if a valid cookie ticket is detected. For more details, see

IPassportManager2::LogoTag2 in the Passport SDK documentation. See Passport documentation for IPassportManager2::LoginUser . See Passport documentation for IPassportManager2::LoginUser .

LogoTag2

[C#] public string LogoTag2(string strReturnUrl, int iTimeWindow, int iForceLogin, string strCoBrandedArgs, int iLangID, int iSecure, string strNameSpace, int iKPP, int iUseSecureAuth);
[C++] public: String\* LogoTag2(String\* strReturnUrl, int iTimeWindow, int iForceLogin, String\* strCoBrandedArgs, int iLangID, int iSecure, String\* strNameSpace, int iKPP, int iUseSecureAuth);
[VB] Public Function LogoTag2(ByVal strReturnUrl As String, ByVal iTimeWindow As Integer, ByVal iForceLogin As Integer, ByVal strCoBrandedArgs As String, ByVal iLangID As Integer, ByVal iSecure As Integer, ByVal strNameSpace As String, ByVal iKPP As Integer, ByVal iUseSecureAuth As Integer) As String
[JScript] public function LogoTag2(strReturnUrl: String, iTimeWindow: int, iForceLogin: int, strCoBrandedArgs: String, iLangID: int, iSecure: int, strNameSpace: String, iKPP: int, iUseSecureAuth: int): String;

lee **@**hayes plic 509+324-9256 423 *MS1-863US.APP* 

## Description

Returns an HTML fragment containing an < img > tag for a Passport link.

This is based on the current state of the identity (already logged on, and so on).

The link image displays either Sign In if no valid ticket cookie is detected, or Sign Out if a valid cookie ticket is detected. For more details, see IPassportManager2::LogoTag2 in the Passport SDK documentation. See Passport documentation for IPassportManager2::LoginUser . See Passport

SignOut

[C#] public static void SignOut(string strSignOutDotGifFileName);
[C++] public: static void SignOut(String\* strSignOutDotGifFileName);
[VB] Public Shared Sub SignOut(ByVal strSignOutDotGifFileName As String)
[JScript] public static function SignOut(strSignOutDotGifFileName : String);

## Description

Logs off the given Passport member from the current session.

For more information, see the Passport SDK documentation . The *SignOut* image to be used.

1	UrlAuthorizationModule class (System.Web.Security)
2	ToString
3	
4	
5	Description
6	Provides URL-based authorization services for allowing or denying access
7	to specified resources. This class cannot be inherited.
8	UrlAuthorizationModule
9	Example Syntax:
10	ToString
11	
12	[C#] public UrlAuthorizationModule();
13	[C++] public: UrlAuthorizationModule();
14	[VB] Public Sub New()
15	[JScript] public function UrlAuthorizationModule();
16	Dispose
17	
18	[C#] public void Dispose();
19	[C++] public:sealed void Dispose();
20	[VB] NotOverridable Public Sub Dispose()
21	[JScript] public function Dispose();
22	
23	Description
24	Called by the HTTP runtime to dispose of the module.
25	

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Call System. Web. Security. Url Authorization Module. Dispose when you are finished using the System. Web. Security. Url Authorization Module. The System. Web. Security. Url Authorization Module. Dispose method leaves the System. Web. Security. Url Authorization Module in an unusable state. After calling System.Web.Security.UrlAuthorizationModule.Dispose, you must release all references to the System. Web. Security. Url Authorization Module so the memory it occupid can be reclaimed by garbage collection. Init [C#] public void Init(HttpApplication app); [C++] public: sealed void Init(HttpApplication\* app); [VB] NotOverridable Public Sub Init(ByVal app As HttpApplication) [JScript] public function Init(app: HttpApplication); Description Initializes the module. The Http application. WindowsAuthenticationEventArgs class (System.Web.Security) **ToString** 

Description

Provides data for the WindowsAuthentication event.

This is the event argument passed to the

WindowsAuthentication\_OnAuthenticate event handler. Contains a

WindowsIdentity object and the IPrincipal object used for the context.

	1	WindowsAuthenticationEventArgs
	2	Example Syntax:
	3	ToString
	4	
	5	[C#] public WindowsAuthenticationEventArgs(WindowsIdentity identity,
	6	HttpContext context);
	7	[C++] public: WindowsAuthenticationEventArgs(WindowsIdentity* identity,
	8	HttpContext* context);
	9	[VB] Public Sub New(ByVal identity As WindowsIdentity, ByVal context As
	10	HttpContext)
4. 10 10 10 10 10 10 10 10 10 10 10 10 10	11	[JScript] public function WindowsAuthenticationEventArgs(identity:
	12	WindowsIdentity, context: HttpContext);
	13	
1)	14	Description
25 - 25 - 25 - 25 - 25 - 25 - 25 - 25 -	15	Initializes a newly created instance of the
	16	WindowsAuthenticationEventArgs Class. The windows identity object. The
1 22	17	context for the event.
	18	Context
	19	ToString
	20	
	21	[C#] public HttpContext Context {get;}
	22	[C++] public:property HttpContext* get_Context();
	23	[VB] Public ReadOnly Property Context As HttpContext
	24	[JScript] public function get Context() : HttpContext;
		II

```
Description
                                                               The HttpContext intrinsic (provides access to Request, Response, and User
                      3
                                    objects).
                                                               Identity
                      5
                                                               ToString
                       6
                       7
                                     [C#] public WindowsIdentity Identity {get;}
                       8
                                     [C++] public: property WindowsIdentity* get_Identity();
                       9
                                     [VB] Public ReadOnly Property Identity As WindowsIdentity
The state and the state of the 
                    10
                                      [JScript] public function get Identity(): WindowsIdentity;
                    11
                    12
                                      Description
                      13
                                                                  An authenticated Windows identity.
                      14
                                                                  User
                      15
                                                                  ToString
                      16
                      17
                                        [C#] public IPrincipal User {get; set;}
                       18
                                        [C++] public: __property IPrincipal* get_User();public: __property void
                       19
                                        set User(IPrincipal*);
                       20
                                        [VB] Public Property User As IPrincipal
                       21
                                        [JScript] public function get User(): IPrincipal; public function set
                       22
                                        User(IPrincipal);
                       23
                       24
                                         Description
```

IPrincipal object to be associated with the request. The user object should be attached to the contextIf User is non null and Context. User is null, the WindowsAuthenticationModule will initialize Context. User with WindowsAuthenticationEventArgs. User.

WindowsAuthenticationEventHandler delegate (System. Web. Security)

ToString

Description

Represents the method that handles the

WindowsAuthentication\_OnAuthenticate event of a

WindowsAuthenticationModule . The source of the event. A

WindowsAuthenticationEventArg that contains the event data.

When you create a

System.Web.Security.WindowsAuthenticationEventHandler delegate, you identify the method to handle the event. To associate the event with your EventHandler, add an instance of the delegate to the event. The EventHandler is called whenever the event occurs, unless you remove the delegate. For more information about EventHandler delegates, see .

WindowsAuthenticationModule class (System.Web.Security)
ToString

Description

Enables ASP.NET applications to use Windows/IIS authentication. This 1 class cannot be inherited. 2 WindowsAuthenticationModule 3 Example Syntax: **ToString** 5 6 [C#] public WindowsAuthenticationModule(); 7 [C++] public: WindowsAuthenticationModule(); 8 [VB] Public Sub New() 9 [JScript] public function WindowsAuthenticationModule(); 10 **ToString** 11 12 [C#] public event WindowsAuthenticationEventHandler Authenticate; 13 [C++] public: \_\_event WindowsAuthenticationEventHandler\* Authenticate; 14 [VB] Public Event Authenticate As WindowsAuthenticationEventHandler 15 16 Description 17 Raised during authentication. This is a Global.asax event that must be 18 named PassportAuthentication\_OnAuthenticate event. It is used primarily to 19 attach a custom IPrincipal object to the context. 20 For more information about handling events, see . 21 Dispose 22 23 [C#] public void Dispose(); 24 [C++] public: \_\_sealed void Dispose();

	1	[VB] NotOverridable Public Sub Dispose()
	2	[JScript] public function Dispose();
	3	
	4	Description
	5	Disposes of the module derived from IHttpModule when called by the
	6	HttpRuntime .
	7	Call System.Web.Security.WindowsAuthenticationModule.Dispose
	8	when you are finished using the
	9	System.Web.Security.WindowsAuthenticationModule . The
	10	System.Web.Security.WindowsAuthenticationModule.Dispose method leaves
and the r	11	the System. Web. Security. Windows Authentication Module in an unusable state
	12	After calling System. Web. Security. Windows Authentication Module. Dispose,
	13	you must release all references to the
	14	System.Web.Security.WindowsAuthenticationModule so the memory it
	15	occupied can be reclaimed by garbage collection.
	16	Init
7	17	
	18	[C#] public void Init(HttpApplication app);
	19	[C++] public:sealed void Init(HttpApplication* app);
	20	[VB] NotOverridable Public Sub Init(ByVal app As HttpApplication)
	21	[JScript] public function Init(app: HttpApplication);
	22	•
	23	Description
	24	Initializes the module derived from IHttpModule when cal
	25	

## System.Web.Services

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

The **System.Web.Services** namespace consists of the classes that enable you to build and use Web Services. A Web Service is a programmable entity residing on a Web Server exposed via standard Internet protocols.

WebMethodAttribute class (System.Web.Services)

## Description

Adding this attribute to a method within an ASP.NET Web Service makes the method callable from remote Web clients. This class cannot be inherited.

Methods within a class that have this attribute set are called Web Service methods. The method and class must be public and running inside an ASP.NET Web application.

Constructors:

WebMethodAttribute

Example Syntax:

[C#] public WebMethodAttribute();

[C++] public: WebMethodAttribute();

[VB] Public Sub New()

[JScript] public function WebMethodAttribute(); Initializes a new instance of the

	1	System.Web.Services.WebMethodAttribute class.
	2	
	3	Description
	4	Initializes a new instance of the
	5	System.Web.Services.WebMethodAttribute class.
	6	WebMethodAttribute
	7	Example Syntax:
	8	
	9	[C#] public WebMethodAttribute(bool enableSession);
A B B B B B B B B B B B B B B B B B B B	10	[C++] public: WebMethodAttribute(bool enableSession);
	11	[VB] Public Sub New(ByVal enableSession As Boolean)
3: 1:1 2:1	12	[JScript] public function WebMethodAttribute(enableSession : Boolean);
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	13	
	14	Description
10 10 10 10 10 10 10 10 10 10 10 10 10 1	15	Initializes a new instance of the
	16	System.Web.Services.WebMethodAttribute class. Initializes whether session
Ž.	17	state is enabled for the Web Service method.
	18	WebMethodAttribute
	19	Example Syntax:
	20	
	21	[C#] public WebMethodAttribute(bool enableSession, TransactionOption
	22	transactionOption);
	23	[C++] public: WebMethodAttribute(bool enableSession, TransactionOption
	24	transactionOption);
	25	[VB] Public Sub New(ByVal enableSession As Boolean, ByVal transactionOption

As TransactionOption) [JScript] public function WebMethodAttribute(enableSession : Boolean, 2 transactionOption: TransactionOption); 3 4 Description 5 Initializes a new instance of the 6 System. Web. Services. WebMethod Attribute class. Initializes whether session 7 state is enabled for the Web Service method. Initializes the transaction support of a 8 Web Service method. 9 WebMethodAttribute 10 Example Syntax: 11 12 [C#] public WebMethodAttribute(bool enableSession, TransactionOption 13 transactionOption, int cacheDuration); 14 [C++] public: WebMethodAttribute(bool enableSession, TransactionOption 15 transactionOption, int cacheDuration); 16 [VB] Public Sub New(ByVal enableSession As Boolean, ByVal transactionOption 17 As TransactionOption, ByVal cacheDuration As Integer) 18 [JScript] public function WebMethodAttribute(enableSession : Boolean, 19 transactionOption: TransactionOption, cacheDuration: int); 20 21 Description 22 Initializes a new instance of the 23 System. Web. Services. WebMethod Attribute class. Initializes whether session 24

	40 40 40 40 40 70 40 10 10 40 40 40 40 40 40 40 40 40 40 40 40 40
;	2
l,	S. Carrie
	300
;	2
į,	in the
Ļ	7
1	3
ij	
ľ,	35
;	No.
	il e
:	* 12 mm.
;	13.F
	75 21
	: :

state is enabled for the Web Service method. Initializes the transaction support of a Web Service method. Initializes the number of seconds the response is cached. WebMethodAttribute

Example Syntax:

2

3

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] public WebMethodAttribute(bool enableSession, TransactionOption transactionOption, int cacheDuration, bool bufferResponse);

[C++] public: WebMethodAttribute(bool enableSession, TransactionOption transactionOption, int cacheDuration, bool bufferResponse);

[VB] Public Sub New(ByVal enableSession As Boolean, ByVal transactionOption As TransactionOption, ByVal cacheDuration As Integer, ByVal bufferResponse As Boolean)

[JScript] public function WebMethodAttribute(enableSession: Boolean, transactionOption: TransactionOption, cacheDuration: int, bufferResponse: Boolean);

Description

Initializes a new instance of the

System. Web. Services. WebMethod Attribute class. Initializes whether session state is enabled for the Web Service method. Initializes the transaction support of a Web Service method. Initializes the number of seconds the response is cached. Initializes whether the response for this request is buffered.

Properties:

BufferResponse

435 MS1-863US.APP

[C#] public bool BufferResponse {get; set;}

[C++] public: \_\_property bool get\_BufferResponse();public: \_\_property void set\_BufferResponse(bool);

[VB] Public Property BufferResponse As Boolean

[JScript] public function get BufferResponse(): Boolean;public function set BufferResponse(Boolean);

Description

Gets or sets whether the response for this request is buffered.

Setting System.Web.Services.WebMethodAttribute.BufferResponse to true, serializes the response of the Web Service method into a memory buffer until either the response is completely serialized or the buffer is full. Once the response is buffered, it is returned to the Web Service client over the network. When System.Web.Services.WebMethodAttribute.BufferResponse is false, the response to the Web Service method is sent back to the client as it is serialized. In general, you only want to set

System.Web.Services.WebMethodAttribute.BufferResponse to false, if it is known that a Web Service method returns large amounts of data to the client. For smaller amounts of data, Web Service performance is better with

 $System. Web. Services. WebMethod Attribute. Buffer Response \ to \ true \ .$ 

CacheDuration

[C#] public int CacheDuration {get; set;}

[C++] public: \_\_property int get\_CacheDuration();public: \_\_property void

17

19

20

21

22

23

24

25

set CacheDuration(int); [VB] Public Property CacheDuration As Integer [JScript] public function get CacheDuration(): int;public function set CacheDuration(int); 5 Description Gets or sets the number of seconds the response should be held in the 7 cache. When caching is enabled requests and responses are held in memory on the 9 server for at least the cache duration so caution must be used if you expect 10 requests or responses to be very large or you expect requests to vary widely. 11 Description 12 13 [C#] public string Description {get; set;} 14 15

[C#] public string Description {get; set;}

[C++] public: \_\_property String\* get\_Description();public: \_\_property void set\_Description(String\*);

[VB] Public Property Description As String

[JScript] public function get Description() : String;public function set Description(String);

## Description

A descriptive message describing the Web Service method.

The descriptive message is displayed to prospective consumers of the Web Service when description documents for the Web Service are generated, such as the Service Description and the Service help page.

2

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

		$\sim$		
Ena	hl.	$\sim$	ACC1	$\alpha$ n

[C#] public bool EnableSession {get; set;}

[C++] public: \_\_property bool get\_EnableSession();public: \_\_property void set EnableSession(bool);

[VB] Public Property EnableSession As Boolean

[JScript] public function get EnableSession() : Boolean; public function set EnableSession(Boolean);

## Description

Indicates whether session state is enabled for a Web Service method.

In order to store session state in the ASP.NET

System.Web.SessionState.HttpSessionState object, the Web Service must inherit from System.Web.Services.WebService and a

System.Web.Services.WebMethodAttribute applied to the Web Service method, setting the System.Web.Services.WebMethodAttribute.EnableSession property to true. If session state is not needed for a Web Service method, then disabling it may improve performance.

MessageName

[C#] public string MessageName {get; set;}

[C++] public: \_\_property String\* get\_MessageName();public: \_\_property void set MessageName(String\*);

[VB] Public Property MessageName As String

[JScript] public function get MessageName(): String; public function set

MessageName(String);

2

Description

4

returned from a Web Service method.

6

10

11 12

13 14 15

16

17

18

20

19

21 22

23

24 25 The name used for the Web Service method in the data passed to and

The System.Web.Services.WebMethodAttribute.MessageName property can be used to alias method or property names. The most common use of the System.Web.Services.WebMethodAttribute.MessageName property will be to uniquely identify polymorphic methods. By default,

System.Web.Services.WebMethodAttribute.MessageName is set to the name of the Web Service method. Therefore, if a Web Service contains two or more Web Service methods with the same name, you can uniquely identify the individual Web Service methods by setting the

System. Web. Services. WebMethod Attribute. Message Name to a name unique within the Web Service, without changing the name of the actual method name in code.

**TransactionOption** 

[C#] public TransactionOption TransactionOption {get; set;}

[C++] public: property TransactionOption get TransactionOption();public:

property void set TransactionOption(TransactionOption);

[VB] Public Property TransactionOption As TransactionOption

[JScript] public function get TransactionOption(): TransactionOption; public

function set TransactionOption(TransactionOption);

Description

Indicates the transaction support of a Web Service method.

Web Service methods can only participate as the root object in a transaction, due to the stateless nature of the HTTP protocol. Web Service methods can invoke COM objects that participate in the same transaction as the Web Service method, if the COM object is marked to run within a transaction in the Component Services administrative tool. If a Web Service method, with a System.Web.Services.WebMethodAttribute.TransactionOption property of Required or RequiresNew invokes another Web Service method with a System.Web.Services.WebMethodAttribute.TransactionOption property of Required or RequiresNew, each Web Service method participates in their own transaction, because a Web Service method can only act as the root object in a transaction.

TypeId

Methods:

WebService class (System.Web.Services)

**ToString** 

Description

Defines the optional base class for Web Services, which provides direct access to common ASP.NET objects, like those for application and session state.

If you don't need access to the common ASP.NET objects, you can still create a Web Service without deriving from **System.Web.Services.WebService**.

Additional ASP.NET objects can be accessed through  $System. Web. Services. Web Service. Context\ .$ 2 WebService 3 Example Syntax: 4 **ToString** 5 6 [C#] public WebService(); [C++] public: WebService(); 8 [VB] Public Sub New() 9 [JScript] public function WebService(); 10 Application 11 **ToString** 12 13 [C#] public HttpApplicationState Application {get;} 14 [C++] public: \_\_property HttpApplicationState\* get\_Application(); 15 [VB] Public ReadOnly Property Application As HttpApplicationState 16 [JScript] public function get Application(): HttpApplicationState; 17 18 Description 19 Gets the application object for the current HTTP request. 20 Web Services can utilize both application state and session state. 21 Application state is maintained across all sessions accessing a Web Service 22 regardless of whether session state is turned off for a method(by using the 23 System.Web.Services.WebMethodAttribute.EnableSession property of the 24  $System. Web. Services. WebMethod Attribute\ ).$ 

7

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Container

Context

**ToString** 

Description

Gets the ASP.NET **System.Web.HttpContext** for the current request, which encapsulates all HTTP-specific context used by the HTTP server to process Web requests.

If a Web Service method is one-way, then the

System.Web.Services.WebService.Context property is always null . A Web Service is one-way when a

 $System. Web. Services. Protocols. So ap Document Method Attribute \ or \ an experimentary and the protocol of the protocol o$ 

System.Web.Services.Protocols.SoapRpcMethodAttribute is applied to a Web Service method and the

System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay property is set to true.

DesignMode

**Events** 

Server

**ToString** 

Description

Gets the System. Web. HttpServerUtility for the current request.

1	The System.Web.HttpServerUtility class provides several methods that
2	can be used in the processing of Web requests, including
3	System.Web.HttpServerUtility.CreateObject(System.String) (for instantiating
4	COM objects).
5	Session
6	ToString
7	
8	[C#] public HttpSessionState Session {get;}
9	[C++] public:property HttpSessionState* get_Session();
10	[VB] Public ReadOnly Property Session As HttpSessionState
11	[JScript] public function get Session(): HttpSessionState;
12	
13	Description
14	Gets the System.Web.SessionState.HttpSessionState instance for the
15	current request.
16	If a Web Service method is one-way, then the
17	System.Web.Services.WebService.Session property is always null. A Web
18	Service is one-way when a
19	System.Web.Services.Protocols.SoapDocumentMethodAttribute or
20	System.Web.Services.Protocols.SoapRpcMethodAttribute is applied to a Web
21	Service method and the
22	System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay
23	property is set to <b>true</b> .
24	Site .
25	User

**ToString** 

Description

Gets the ASP.NET server **System.Web.HttpContext.User** object. Can be used to authenticate whether a user is authorized to execute the request.

Both Internet Information Services (IIS) and the .NET Framework need to be configured for authentication in order for the

System. Web. Services. Web Service. User property to be meaningful.

Authentication is the process of accepting credentials from a user and validating those credentials against some authority. If the credentials are valid, you have an authenticated identity. Authentication in the .NET Framework is configured by placing entries in the web.config file. Placing the XML in the following code into a web.config file sets the authentication mode to Windows.

WebServiceAttribute class (System.Web.Services)
ToString

Description

Used to add additional information to a Web Service, such as a string describing its functionality.

The System.Web.Services.WebServiceAttribute is not required for a
Web Service to be published and executed. You can use the

WebServiceAttribute to specify a name for the Web Service that is not restricted

by the rules for a Common Language Runtime identifier, which is used in the Service Description and the Service help page for the Web Service. 2 **ToString** 3 [C#] public const string DefaultNamespace; 5 [C++] public: const String\* DefaultNamespace; [VB] Public Const DefaultNamespace As String 7 [JScript] public var DefaultNamespace : String; 8 9 Description 10 The default value for the 11 System.Web.Services.WebServiceAttribute.Namespace property. This field is 12 constant. 13 WebServiceAttribute 14 Example Syntax: 15 **ToString** 16 17 [C#] public WebServiceAttribute(); 18 [C++] public: WebServiceAttribute(); 19 [VB] Public Sub New() 20 [JScript] public function WebServiceAttribute(); 21 22 Description 23 Initializes a new instance of the 24  ${\bf System. Web. Services. Web Service Attribute\ class.}$ 

1	Description
2	ToString
3	
4	[C#] public string Description {get; set;}
5	[C++] public:property String* get_Description();public:property void
6	set_Description(String*);
7	[VB] Public Property Description As String
8	[JScript] public function get Description(): String; public function set
9	Description(String);
10	
11	Description
12	A descriptive message for the Web Service.
13	The descriptive message is displayed to prospective consumers of the Web
14	Service when description documents for the Web Service are generated, such as
15	the Service Description and the Service help page.
16	Name
17	ToString
18	
19	[C#] public string Name {get; set;}
20	[C++] public:property String* get_Name();public:property void
21	set_Name(String*);
22	[VB] Public Property Name As String
23	[JScript] public function get Name(): String;public function set Name(String);
24	
25	Description

Gets or sets the name of the ASP.NET Web Service.

The Service Description is generated when a user navigates to the URL for the Web Service and supplies a query string of ?WSDL. Within the Service Description, the **System.Web.Services.WebServiceAttribute.Name** property identifies the local part of the XML qualified name for the Web Service. The **Name** property is also used to display the name of the Web Service on the Service help page. The Service help page is displayed when a prospective consumer navigates to the .asmx page for the Web Service without specifying a Web Service method name and its parameters.

Namespace

**ToString** 

[C#] public string Namespace {get; set;}

[C++] public: \_\_property String\* get\_Namespace();public: \_\_property void set Namespace(String\*);

[VB] Public Property Namespace As String

[JScript] public function get Namespace(): String; public function set Namespace(String);

Description

Gets or sets the default XML namespace to use for the Web Service.

XML namespaces offer a way to create names in an XML document that are identified by a Uniform Resource Identifier (URI). By using XML namespaces you can uniquely identify elements or attributes in a XML document. The Service

[VB] Public Sub New()

25

Description for a Web Service is defined in XML, specifically in Web Service 1 Description Language (WSDL). 2 TypeId 3 WebServiceBindingAttribute class (System.Web.Services) **ToString** 5 7 Description 8 Declares the binding one or more Web Service methods implemented 9 within the class implementing the Web Service. This class cannot be inherited. 10 A binding, as defined by Web Services Description Language (WSDL), is 11 similar to an interface, in that it defines a concrete set of operations. Each Web 12 Service method is an operation within a particular binding. Web Service methods 13 are members of either the default binding for a Web Service or in a binding 14 specified within a System. Web. Services. Web Service Binding Attribute applied 15 to a class implementing a Web Service. A Web Service can implement multiple 16 bindings, by applying multiple 17 System.Web.Services.WebServiceBindingAttribute attributes to a Web Service. 18 WebServiceBindingAttribute 19 Example Syntax: 20 **ToString** 21 22 [C#] public WebServiceBindingAttribute(); 23 [C++] public: WebServiceBindingAttribute(); 24

2

3

4

5

7

8

9

10

11

12

15

16

17

18

19

20

21

22

23

24

25

[JScript] public function WebServiceBindingAttribute(); Initializes a new instance of the System. Web. Services. Web Service Binding Attribute class. Description Initializes a new instance of the  ${\bf System. Web. Services. Web Service Binding Attribute\ class.}$ WebServiceBindingAttribute Example Syntax: ToString [C#] public WebServiceBindingAttribute(string name); [C++] public: WebServiceBindingAttribute(String\* name); [VB] Public Sub New(ByVal name As String) [JScript] public function WebServiceBindingAttribute(name : String); Description Initializes a new instance of the System. Web. Services. Web Service Binding Attribute class setting the name of the binding the Web Service method is implementing. This constructor is used to specify a name for a binding defined in the Web Service it is applied to and is a member of the default namespace. The name of the binding a Web Service method is implementing an operation for. Sets the

WebServiceBindingAttribute

Example Syntax:

System. Web. Services. Web Service Binding Attribute. Name property.

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

[C#] public WebServiceBindingAttribute(string name, string ns);

[C++] public: WebServiceBindingAttribute(String\* name, String\* ns);

[VB] Public Sub New(ByVal name As String, ByVal ns As String)

[JScript] public function WebServiceBindingAttribute(name : String, ns : String);

Description

Initializes a new instance of the

System.Web.Services.WebServiceBindingAttribute class.

This constructor is used to specify a name for a binding defined in the Web Service it is applied to that and is a member of the supplied namespace. The name of the binding a Web Service method is implementing an operation for. Sets the **System.Web.Services.WebServiceBindingAttribute.Name** property. The namespace associated with the binding. Sets the

 $System. Web. Services. WebService Binding Attribute. Names pace \ property.$ 

WebServiceBindingAttribute

Example Syntax:

**ToString** 

20

21

22

23

24

[C#] public WebServiceBindingAttribute(string name, string ns, string location);

[C++] public: WebServiceBindingAttribute(String\* name, String\* ns, String\*

location);

[VB] Public Sub New(ByVal name As String, ByVal ns As String, ByVal location

|As String|

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[JScript] public function WebServiceBindingAttribute(name: String, ns: String, location: String); Description Initializes a new instance of the System. Web. Services. Web Service Binding Attribute class. This constructor is used to specify a name for a binding not defined in the Web Service it is applied to that and is a member of the supplied namespace. The name of the binding a Web Service method is implementing an operation for. Sets the System. Web. Services. Web Service Binding Attribute. Name property. The namespace associated with the binding. Sets the System.Web.Services.WebServiceBindingAttribute.Namespace property. The location where the binding is defined. Location **ToString** [C#] public string Location {get; set;} [C++] public: property String\* get Location(); public: property void set Location(String\*); [VB] Public Property Location As String [JScript] public function get Location(): String; public function set Location(String); Description

Gets or sets the location where the binding is defined.

15

16

17

18

19

20

21

22

23

24

25

Name **ToString** 

2

3

5

6

7

8

9

[C#] public string Name {get; set;}

[C++] public: property String\* get Name();pub

# **System.Web.Services.Configuration**

Description

WebServicesConfigurationSectionHandler class (System.Web.Services.Configuration)

Description

The configuration section handler for the webServices section of the Config. Web configuration file. The section handler participates in the resolution of configuration settings within the webServices portion of a Config.Web.

Constructors:

WebServicesConfigurationSectionHandler

Example Syntax:

[C#] public WebServicesConfigurationSectionHandler();

[C++] public: WebServicesConfigurationSectionHandler();
[VB] Public Sub New()

[JScript] public function WebServicesConfigurationSectionHandler();

Methods:

Create

[C#] public object Create(object parent, object configContext, XmlNode section);
[C++] public: \_\_sealed Object\* Create(Object\* parent, Object\* configContext, XmlNode\* section);
[VB] NotOverridable Public Function Create(ByVal parent As Object, ByVal configContext As Object, ByVal section As XmlNode) As Object
[JScript] public function Create(parent : Object, configContext : Object, section : XmlNode) : Object;

# Description

Parses the configuration settings for the webServices portion of a Web.config configuration file to populate the values of a WebServicesConfiguration object and returning it. Reference to the "default" value provided by the parent IConfigurationSectionHandler. Provides access to the raw XML contents within a configuration file. The virtual path for which the configuration section handler should compute values.

XmlFormatExtensionAttribute class (System.Web.Services.Configuration)
ToString

	- 11	
	1	
	2	
	3	Description
	4	
	5	XmlFormatExtensionAttribute
	6	Example Syntax:
	7	ToString
	8	
	9	[C#] public XmlFormatExtensionAttribute();
18 mg	10	[C++] public: XmlFormatExtensionAttribute();
7 T T T T T T T T T T T T T T T T T T T	11	[VB] Public Sub New()
A de de la militar de la marca del la marca de la marc	12	[JScript] public function XmlFormatExtensionAttribute();
hii an an	13	
	14	Description
	15	
	16	XmlFormatExtensionAttribute
2: }:	17	Example Syntax:
	18	ToString
	19	
	20	[C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type
	21	extensionPoint1);
	22	[C++] public: XmlFormatExtensionAttribute(String* elementName, String* ns,
	23	Type* extensionPoint1);
	24	[VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal
	25	extensionPoint1 As Type)

	1	[JScript] public function XmlFormatExtensionAttribute(elementName : String, ns
	2	: String, extensionPoint1 : Type);
	3	XmlFormatExtensionAttribute
	4	Example Syntax:
	5	ToString
	6	
	7	[C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type[]
	8	extensionPoints);
	9	[C++] public: XmlFormatExtensionAttribute(String* elementName, String* ns,
	10	Type* extensionPoints[]);
H. H. M. S. Salling Breen, Mrs. State Stat	11	[VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal
	12	extensionPoints() As Type)
	13	[JScript] public function XmlFormatExtensionAttribute(elementName : String, ns
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	14	: String, extensionPoints : Type[]);
; ====	15	
	16	Description
***	17	
	18	XmlFormatExtensionAttribute
	19	Example Syntax:
	20	ToString
	21	
	22	[C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type
	23	extensionPoint1, Type extensionPoint2);
	24	[C++] public: XmlFormatExtensionAttribute(String* elementName, String* ns,
	25	Type* extensionPoint1, Type* extensionPoint2);

[VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal extensionPoint1 As Type, ByVal extensionPoint2 As Type) 2 [JScript] public function XmlFormatExtensionAttribute(elementName: String, ns 3 : String, extensionPoint1 : Type, extensionPoint2 : Type); **XmlFormatExtensionAttribute** 5 Example Syntax: **ToString** 7 8 [C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type extensionPoint1, Type extensionPoint2, Type extensionPoint3); [C++] public: XmlFormatExtensionAttribute(String\* elementName, String\* ns, Type\* extensionPoint1, Type\* extensionPoint2, Type\* extensionPoint3); 12 [VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal 13 extensionPoint1 As Type, ByVal extensionPoint2 As Type, ByVal 14 extensionPoint3 As Type) 15 [JScript] public function XmlFormatExtensionAttribute(elementName: String, ns 16 : String, extensionPoint1 : Type, extensionPoint2 : Type, extensionPoint3 : Type); 17 XmlFormatExtensionAttribute 18 Example Syntax: 19 **ToString** 20 21 [C#] public XmlFormatExtensionAttribute(string elementName, string ns, Type 22 extensionPoint1, Type extensionPoint2, Type extensionPoint3, Type 23 extensionPoint4); 24 [C++] public: XmlFormatExtensionAttribute(String\* elementName, String\* ns,

```
Type* extensionPoint1, Type* extensionPoint2, Type* extensionPoint3, Type*
         extensionPoint4);
     2
         [VB] Public Sub New(ByVal elementName As String, ByVal ns As String, ByVal
         extensionPoint1 As Type, ByVal extensionPoint2 As Type, ByVal
         extensionPoint3 As Type, ByVal extensionPoint4 As Type)
     5
         [JScript] public function XmlFormatExtensionAttribute(elementName: String, ns
         : String, extensionPoint1 : Type, extensionPoint2 : Type, extensionPoint3 : Type,
     7
         extensionPoint4: Type);
     8
                Properties:
     9
ElementName
     10
                ToString
     11
     12
         [C#] public string ElementName {get; set;}
    13
[C++] public: property String* get ElementName();public: property void
     14
         set ElementName(String*);
     15
         [VB] Public Property ElementName As String
     16
         [JScript] public function get ElementName(): String; public function set
     17
         ElementName(String);
     18
     19
         Description
     20
     21
                ExtensionPoints
     22
                ToString
     23
     24
         [C#] public Type[] ExtensionPoints {get; set;}
```

lee@hayes pilc 509-324-9256 457 MS1-863US.APP

```
[C++] public: property Type* get_ExtensionPoints();public: __property void
                                 set ExtensionPoints(Type*[]);
                                 [VB] Public Property ExtensionPoints As Type ()
                                  [JScript] public function get ExtensionPoints(): Type[];public function set
                                  ExtensionPoints(Type[]);
                    6
                                  Description
                    8
                                                            Namespace
The state of the s
                                                            ToString
                 11
                                  [C#] public string Namespace {get; set;}
                 12
                                  [C++] public: property String* get Namespace(); public: property void
                 13
set Namespace(String*);
                 14
                                  [VB] Public Property Namespace As String
                 15
                                  [JScript] public function get Namespace(): String; public function set
                 16
                                  Namespace(String);
                 17
                 18
                                  Description
                 19
                 20
                                                            TypeId
                 21
                                                            XmlFormatExtensionPointAttribute class
                 22
                                  (System. Web. Services. Configuration)
                 23
                                                            ToString
                 24
                 25
```

	1	
	2	
	3	Description
	4	
	5	XmlFormatExtensionPointAttribute
	6	Example Syntax:
	7	ToString
	8	
	9	[C#] public XmlFormatExtensionPointAttribute(string memberName);
	10	[C++] public: XmlFormatExtensionPointAttribute(String* memberName);
	11	[VB] Public Sub New(ByVal memberName As String)
1	12	[JScript] public function XmlFormatExtensionPointAttribute(memberName
# # # # # # # # # # # # # # # # # # #	13	String);
## ## ## ## ## ## ## ## ## ## ## ## ##	14	
: क्रेंग	15	Description
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	16	
de	17	AllowElements
	18	ToString
	19	
	20	[C#] public bool AllowElements {get; set;}
	21	[C++] public:property bool get_AllowElements();public:property void
	22	set_AllowElements(bool);
	23	[VB] Public Property AllowElements As Boolean
	24	[JScript] public function get AllowElements(): Boolean; public function set
	25	AllowElements(Boolean);
		•

1	
}	Description
	2 oser quen
	Monthanland
4	MemberName
5	ToString
6	
7	[C#] public string MemberName {get; set;}
8	[C++] public:property String* get_MemberName();public:property void
9	set_MemberName(String*);
10	[VB] Public Property MemberName As String
11	[JScript] public function get MemberName(): String; public function set
12	MemberName(String);
13	·
14	Description
15	
16	TypeId
17	XmlFormatExtensionPrefixAttribute class
18	(System.Web.Services.Configuration)
19	ToString
20	
21	
22	Description
23	
24	XmlFormatExtensionPrefixAttribute
25	Example Syntax:
	6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24

	1	ToString
	2	
	3	[C#] public XmlFormatExtensionPrefixAttribute();
	4	[C++] public: XmlFormatExtensionPrefixAttribute();
	5	[VB] Public Sub New()
	6	[JScript] public function XmlFormatExtensionPrefixAttribute();
	7	
	8	Description
A 18 18 18	9	
	10	XmlFormatExtensionPrefixAttribute
# # # # # # # # # # # # # # # # # # #	11	Example Syntax:
	12	ToString
# 1 m	13	
	14	[C#] public XmlFormatExtensionPrefixAttribute(string prefix, string ns);
# # 14 14 19 19 19 19 19 19 19 19 19 19 19 19 19	15	[C++] public: XmlFormatExtensionPrefixAttribute(String* prefix, String* ns);
	16	[VB] Public Sub New(ByVal prefix As String, ByVal ns As String)
: # <del>*</del> ;	17	[JScript] public function XmlFormatExtensionPrefixAttribute(prefix: String, ns
	18	String);
	19	
	20	Description
	21	
	22	Namespace
	23	ToString
	24	
	25	[C#] public string Namespace {get; set;}

```
[C++] public: property String* get Namespace(); public: __property void
        set Namespace(String*);
     2
        [VB] Public Property Namespace As String
        [JScript] public function get Namespace(): String; public function set
        Namespace(String);
     5
     6
        Description
     8
               Prefix
     9
ToString
    10
    11
        [C#] public string Prefix {get; set;}
    12
        [C++]
    13
14
* *
    15
    16
         System.Web.Services.Description
    17
```

### Description

The System.Web.Services.Description namespace consists of the classes that enable you to publicly describe a Web Service by using the Web Service Description Language (WSDL). Each class in the System.Web.Services.Description namespace corresponds to a specific element

in the WSDL specification, and the class hierarchy corresponds to the XML

25

18

19

20

21

22

23

24

	1	structure of a valid WSDL document. For more information about WSDL, see the
	2	specification at http://www.w3.org/TR/wsdl/.
	3	Binding class (System.Web.Services.Description)
	4	
	5	
	6	Description
	7	Specifies the concrete data format and protocols used in the Web Service.
	8	This class cannot be inherited.
	9	Constructors:
M. W. W. W. W.	10	Binding
# 10 mm	11	Example Syntax:
	12	
The state of the s	13	[C#] public Binding();
) : #! : #!	14	[C++] public: Binding();
Apr. 48 of Mary Str. 2017. Str. 18 of Mary Str	15	[VB] Public Sub New()
201 201 201	16	[JScript] public function Binding();
400	17	Properties:
	18	Documentation
	19	Extensions
	20	
	21	
	22	Description
	23	Gets the collection of extensibility elements used in the Web Service.
	24	Name

1	
2	[C#] public string Name {get; set;}
3	[C++] public:property String* get_Name();public:property void
4	set_Name(String*);
5	[VB] Public Property Name As String
6	[JScript] public function get Name(): String; public function set Name(String);
7	
8	Description
9	Gets or sets a string value containing the name of the
10	System.Web.Services.Description.Binding.
11	Operations
12	
13	[C#] public OperationBindingCollection Operations {get;}
14	[C++] public:property OperationBindingCollection* get_Operations();
15	[VB] Public ReadOnly Property Operations As OperationBindingCollection
16	[JScript] public function get Operations(): OperationBindingCollection;
17	
18	Description
19	Gets the collection of the specifications for data formats and message
20	protocols used in the action supported by the Web Service.
21	ServiceDescription
22	
23	[C#] public ServiceDescription ServiceDescription {get;}
24	[C++] public:property ServiceDescription* get_ServiceDescription();
25	[VB] Public ReadOnly Property ServiceDescription As ServiceDescription

1	[JScript] public function get ServiceDescription(): ServiceDescription;
2	
3	Description
4	Gets the System. Web. Services. Description. Service Description instance
5	of which the System. Web. Services. Description. Binding is a member.
6	Туре
7	
8	[C#] public XmlQualifiedName Type {get; set;}
9	[C++] public:property XmlQualifiedName* get_Type();public:property
10	<pre>void set_Type(XmlQualifiedName*);</pre>
11	[VB] Public Property Type As XmlQualifiedName
12	[JScript] public function get Type(): XmlQualifiedName; public function set
13	Type(XmlQualifiedName);
14	
15	Description
16	Gets or sets a value representing the XML datatype definitions used by the
17	Web Service.
18	The default implementation is String.Empty.
19	Methods:
20	BindingCollection class (System.Web.Services.Description)
21	ToString
22	
23	
24	Description
25	

Table

25

1 Represents a collection of System. Web. Services. Description. Binding elements supported by the Web Service. This class cannot be inherited. 2 Count 3 InnerList Item 5 **ToString** System. Web. Services. Description. Binding 7 8 Description 9 Gets or sets the value of a System. Web. Services. Description. Binding at 10 the specified zero-based index. The zero-based index of the 11 System. Web. Services. Description. Binding whose value is modified or returned. 12 Item 13 **ToString** 14 15 [C#] public Binding this[string name] {get;} 16 [C++] public: property Binding\* get Item(String\* name); 17 [VB] Public Default ReadOnly Property Item(ByVal name As String) As Binding 18 [JScript] returnValue = BindingCollectionObject.Item(name); 19 20 Description 21 Gets a System. Web. Services. Description. Binding specified by its Name 22 property. A string value representing the name of the Binding returned. 23 List 24

2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	
The state of the s	
The second second	
ii	

A. J.	
The same that their terms that the same that the same the same that the	
LIGHT.	

1	Add
2	
3	[C#] public int Add(Binding binding);
4	[C++] public: int Add(Binding* binding);
5	[VB] Public Function Add(ByVal binding As Binding) As Integer
6	[JScript] public function Add(binding : Binding) : int;
7	
8	Description
9	Adds the specified <b>System.Web.Services.Description.Binding</b> to the end
10	of the System.Web.Services.Description.BindingCollection .
11	Return Value: Returns the zero-based index where binding has been added. The
12	System.Web.Services.Description.Binding to be added to the collection.
13	Contains
14	
15	[C#] public bool Contains(Binding binding);
16	[C++] public: bool Contains(Binding* binding);
17	[VB] Public Function Contains(ByVal binding As Binding) As Boolean
18	[JScript] public function Contains(binding : Binding) : Boolean;
19	
20	Description
21	Gets a value indicating whether the specified
22	System.Web.Services.Description.Binding is a member of the
23	System.Web.Services.Description.BindingCollection.
24	Return Value: true if binding is a member of the

	3
	4
	5
	6
	7
	8
121.	9
100 mm	10
# 1 # 1 # 1	11
Same Amile	12
iller i inner	13
	14
	15
	16
	17
	18
	19
	20
	21

System.Web.Services.Description.BindingCollection; otherwise, false. A	
System.Web.Services.Description.Binding object.	
СоруТо	
[C#] public void CopyTo(Binding[] array, int index):	

[C#] public void CopyTo(Binding[] array, int index);[C++] public: void CopyTo(Binding\* array[], int index);[VB] Public Sub CopyTo(ByVal array() As Binding, ByVal index As Integer)

[JScript] public function CopyTo(array : Binding[], index : int);

# Description

Copies the entire **System.Web.Services.Description.BindingCollection** to a compatible one-dimensional array of type

**System.Web.Services.Description.Binding**, starting at the specified zero-based index of the target array. An array of type

**System.Web.Services.Description.Binding** serving as the destination for the copy action. The zero-based index at which to start placing the copied collection.

GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String

MS1-863LIS APP

[JScript] protected override function GetKey(value : Object) : String;

Description

22

23

24

25

associated with the value passed by reference. An object for which to return the 2 name of the key. 3 IndexOf 5 [C#] public int IndexOf(Binding binding); [C++] public: int IndexOf(Binding\* binding); [VB] Public Function IndexOf(ByVal binding As Binding) As Integer 8 [JScript] public function IndexOf(binding : Binding) : int; 10 Description 11 Searches for the specified System. Web. Services. Description. Binding and 12 returns the zero-based index of the first occurrence within the collection. 13 Return Value: Returns a 32-bit signed integer. A 14 System.Web.Services.Description.Binding object. 15 Insert 16 17 [C#] public void Insert(int index, Binding binding); 18 [C++] public: void Insert(int index, Binding\* binding); 19 [VB] Public Sub Insert(ByVal index As Integer, ByVal binding As Binding) 20 [JScript] public function Insert(index : int, binding : Binding); 21

Returns the name of the System. Web. Services. Description. Binding

Description

22

23

24

25

Adds the specified **System.Web.Services.Description.Binding** to the **System.Web.Services.Description.BindingCollection** at the specified index.

2

3

5

6

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert binding. The System.Web.Services.Description.Binding to be added to the collection.

Remove

[C#] public void Remove(Binding binding);

[C++] public: void Remove(Binding\* binding);

[VB] Public Sub Remove(ByVal binding As Binding)

[JScript] public function Remove(binding: Binding);

Description

Removes the first occurrence of the specified

System.Web.Services.Description.Binding from the

 ${\bf System. Web. Services. Description. Binding Collection}\ .$ 

This method performs a linear search; therefore, the average execution time is proportional to **System.Web.Services.Description.BindingCollection.Count** .

A System. Web. Services. Description. Binding object.

SetParent

[C#] protected override void SetParent(object value, object parent);

[C++] protected: void SetParent(Object\* value, Object\* parent);

[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As

Object)

[JScript] protected override function SetParent(value : Object, parent : Object);

Description

Sets the parent System.Web.Services.Description.ServiceDescription of a member of System.Web.Services.Description.BindingCollection . An object, of type System.Web.Services.Description.Binding , within the collection. The object, of type System.Web.Services.Description.ServiceDescription , to set as the parent.

DocumentableItem class (System.Web.Services.Description)
ToString

#### Description

Represents the abstract base class from which several classes in the **System.Web.Services.Description** namespace are derived.

Several classes are derived from this class, including:

System.Web.Services.Description.BindingSystem.Web.Services.Description.I mportSystem.Web.Services.Description.MessageSystem.Web.Services.Description.OperationSystem.Web.Services.Description.OperationSystem.Web.Services.Description.OperationMessageSystem.Web.Services.Description.PortSystem.Web.Services.Description.PortTypeSystem.Web.Services.Description.ServiceSystem.Web.Services.Description.Types This base class contains only one property, the Documentation property. It represents the text documentation for a specific XML element.

1	DocumentableItem
2	Example Syntax:
3	ToString
4	
5	[C#] protected DocumentableItem();
6	[C++] protected: DocumentableItem();
7	[VB] Protected Sub New()
8	[JScript] protected function DocumentableItem();
9	Documentation
10	ToString
11	
12	[C#] public string Documentation {get; set;}
13	[C++] public:property String* get_Documentation();public:property void
14	set_Documentation(String*);
15	[VB] Public Property Documentation As String
16	[JScript] public function get Documentation(): String; public function set
17	Documentation(String);
18	
19	Description
20	Gets or sets the documentation for the instance of the
21	System.Web.Services.Description.DocumentableItem .
22	In a derived class, this property represents the text comments added to an
23	element of the Web Service. The default implementation is String.Empty.
24	FaultBinding class (System.Web.Services.Description)
25	ToString

	11	
	1	
	2	
	3	Description
	4	Specifies the format for any error messages that might be output as a result
	5	of the operation. This class cannot be inherited.
	6	FaultBinding
	7	Example Syntax:
	8	ToString
:: :::	9	
#. #. #.	10	[C#] public FaultBinding();
A A A A A A A A A A A A A A A A A A A	11	[C++] public: FaultBinding();
	12	[VB] Public Sub New()
	13	[JScript] public function FaultBinding();
	14	Documentation
	15	Extensions
	16	ToString
	17	
	18	
	19	Description
	20	Gets the collection of extensibility elements associated with the
	21	System.Web.Services.Description.FaultBinding.
	22	Name
	23	OperationBinding
	24	FaultBindingCollection class (System.Web.Services.Description)
	25	ToString

	2	
	3	De.
	4	
	5	Sys
	6	inh
	7	
	8	
i.	9	
day tank tank	10	
	11	
	12	
	13	De.
	14	
	15	Sys
	16	ind
	17	Sys
	18	reti
	19	
	20	
	21	

Description

Represents a collection of

**System.Web.Services.Description.FaultBinding** instances. This class cannot be inherited.

Count

InnerList

Item

**ToString** 

System.Web.Services.Description.FaultBinding

Description

Gets or sets the value of a

System.Web.Services.Description.FaultBinding at the specified zero-based

index. The zero-based index of the

**System.Web.Services.Description.FaultBinding** whose value is modified or returned.

Item

**ToString** 

[C#] public FaultBinding this[string name] {get;}

[C++] public: \_\_property FaultBinding\* get\_Item(String\* name);

[VB] Public Default ReadOnly Property Item(ByVal name As String) As

**FaultBinding** 

[JScript] returnValue = FaultBindingCollectionObject.Item(name); 2 Description 3 Gets a System. Web. Services. Description. Fault Binding specified by its System. Web. Services. Description. Fault Binding. Name property. The string value representing the name of the System. Web. Services. Description. Fault Binding returned. 7 List Table Add 10 11 [C#] public int Add(FaultBinding bindingOperationFault); 12 [C++] public: int Add(FaultBinding\* bindingOperationFault); 13 [VB] Public Function Add(ByVal bindingOperationFault As FaultBinding) As 14 Integer 15 [JScript] public function Add(bindingOperationFault : FaultBinding) : int; 16 17 Description 18 Adds the specified System. Web. Services. Description. Fault Binding to the 19 end of the System. Web. Services. Description. Fault Binding Collection. 20 Return Value: Returns the index where bindingOperationFault has been added. 21 The System. Web. Services. Description. Fault Binding to be added to the 22 collection. 23 Contains 24

	1	
	2	[C#] public bool Contains(FaultBinding bindingOperationFault);
	3	[C++] public: bool Contains(FaultBinding* bindingOperationFault);
	4	[VB] Public Function Contains(ByVal bindingOperationFault As FaultBinding)
	5	As Boolean
	6	[JScript] public function Contains(bindingOperationFault : FaultBinding) :
	7	Boolean;
	8	
	9	Description
20. 20. 20. 20. 20.	10	Gets a value indicating whether the specified
Som mail That Aug Bank	11	System. Web. Services. Description. Fault Binding is a member of the
Rang.	12	FaultBindingCollection .
	13	Return Value: <b>true</b> if bindingOperationFault is a member of the
	14	FaultBindingCollection; otherwise, false. A
	15	System.Web.Services.Description.FaultBinding object.
	16	СоруТо
	17	
	18	[C#] public void CopyTo(FaultBinding[] array, int index);
	19	[C++] public: void CopyTo(FaultBinding* array[], int index);
	20	[VB] Public Sub CopyTo(ByVal array() As FaultBinding, ByVal index As
	21	Integer)
	22	[JScript] public function CopyTo(array : FaultBinding[], index : int);
	23	
	24	Description
	25	
	- 1	l e e e e e e e e e e e e e e e e e e e

Anny dans dans

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

2

3

4

5

Copies the entire FaultBindingCollection to a compatible one-dimensional array of type System.Web.Services.Description.FaultBinding, starting at the specified zero-based index of the target array. An array of type

**System.Web.Services.Description.FaultBinding** serving as the destination for the copy action. The zero-based index at which to start placing the copied collection.

GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String [JScript] protected override function GetKey(value : Object) : String;

Description

Returns the name of the **System.Web.Services.Description.FaultBinding** associated with the value passed by reference. An object for which to return the name.

IndexOf

[C#] public int IndexOf(FaultBinding bindingOperationFault);

[C++] public: int IndexOf(FaultBinding\* bindingOperationFault);

[VB] Public Function IndexOf(ByVal bindingOperationFault As FaultBinding) As Integer

[JScript] public function IndexOf(bindingOperationFault : FaultBinding) : int;

Description

3

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Searches for the specified

**System.Web.Services.Description.FaultBinding** and returns the zero-based index of the first occurrence within the collection.

Return Value: Returns a 32-bit signed integer. A

System.Web.Services.Description.FaultBinding instance.

Insert

[C#] public void Insert(int index, FaultBinding bindingOperationFault);

[C++] public: void Insert(int index, FaultBinding\* bindingOperationFault);

[VB] Public Sub Insert(ByVal index As Integer, ByVal bindingOperationFault As FaultBinding)

[JScript] public function Insert(index: int, bindingOperationFault: FaultBinding);

Description

Adds the specified **System.Web.Services.Description.FaultBinding** to the **FaultBindingCollection** at the specified zero-based index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert bindingOperationFault. The System.Web.Services.Description.FaultBinding to be added to the collection.

Remove

1	
2	[C#] public void Remove(FaultBinding bindingOperationFault);
3	[C++] public: void Remove(FaultBinding* bindingOperationFault);
4	[VB] Public Sub Remove(ByVal bindingOperationFault As FaultBinding)
5	[JScript] public function Remove(bindingOperationFault : FaultBinding);
6	
7	Description
8	Removes the first occurrence the specified
9	System.Web.Services.Description.FaultBinding from the
10	FaultBindingCollection .
11	This method performs a linear search; therefore, the average execution time
12	is proportional to
13	${\bf System. Web. Services. Description. Fault Binding Collection. Count}\ .\ The$
14	System.Web.Services.Description.FaultBinding object to be removed from the
15	collection.
16	SetParent
17	
18	[C#] protected override void SetParent(object value, object parent);
19	[C++] protected: void SetParent(Object* value, Object* parent);
20	[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
21	Object)
22	[JScript] protected override function SetParent(value : Object, parent : Object);
23	
24	Description
25	

	1	Sets the parent System. Web. Services. Description. Operation Binding of a
	2	member of the FaultBindingCollection instance. An object, of type
	3	System.Web.Services.Description.FaultBinding, within the
	4	FaultBindingCollection. The object, of type
	5	System.Web.Services.Description.OperationBinding, to set as the parent.
	6	HttpAddressBinding class (System.Web.Services.Description)
	7	ToString
	8	
#4	9	
The state of the s	10	Description
er Sunt Sunt Sunt Sunt Sunt Sunt Sunt Sunt	11	Represents an extensibility element added to a
, dans, man	12	System.Web.Services.Description.Port within a Web Service. This class cannot
	13	be inherited.
	14	This class, through its
	15	System.Web.Services.Description.HttpAddressBinding.Location property,
	16	specifies the base URI for the Web Service. For more information, see .
	17	HttpAddressBinding
	18	Example Syntax:
	19	ToString
	20	
	21	[C#] public HttpAddressBinding();
	22	[C++] public: HttpAddressBinding();
	23	[VB] Public Sub New()
	24	[JScript] public function HttpAddressBinding();
	25	Handled

Location

**ToString** 

Description

3

4

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets a value representing the URL of the .asmx file containing the methods exposed by the Web Service.

Parent

Required

HttpBinding class (System.Web.Services.Description)

**ToString** 

Description

Represents an extensibility element added to a

**System.Web.Services.Description.Binding** within a Web Service. This class cannot be inherited.

This class specifies that information is to be passed by means of HTTP. For more information about specification of protocols for Web Services, see .

**ToString** 

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;

	1	
	2	Description
	3	Specifies the URI (http://schemas.xmlsoap.org/wsdl/http/) for the XML
	4	namespace of the HttpBinding class.
	5	HttpBinding
	6	Example Syntax:
	7	ToString
	8	
2 <b>8</b> 3	9	[C#] public HttpBinding();
The state of the s	10	[C++] public: HttpBinding();
	11	[VB] Public Sub New()
	12	[JScript] public function HttpBinding();
ar ar	13	Handled
Man Anni	14	Parent
F And And He dim And	15	Required
	16	Verb
	17	ToString
	18	
	19	
	20	Description
	21	Gets or sets a value indicating whether the Http request will be made using
	22	the "GET" or "POST" method.
	23	HttpOperationBinding class (System.Web.Services.Description)
	24	ToString
	25	

dDH), dHr. .....

Description

2

3

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

23

24

25

Represents an extensibility element added to an

System. Web. Services. Description. Operation Binding within a Web Service.

This class cannot be inherited.

This class specifies how a Web Service passes information by means of HTTP. For more information, see .

HttpOperationBinding

Example Syntax:

**ToString** 

[C#] public HttpOperationBinding();

[C++] public: HttpOperationBinding();

[VB] Public Sub New()

[JScript] public function HttpOperationBinding();

Handled

Location

**ToString** 

22 Description

Gets or sets a value representing the relative URL, within the WSDL document, of the action supported by the **HttpOperationBinding** .

Parent

	1	Required
	2	HttpUrlEncodedBinding class (System.Web.Services.Description)
	3	ToString
	4	
	5	
	6	Description
	7	Represents an extensibility element added to an
	8	System.Web.Services.Description.InputBinding within a Web Service. This
<b>2</b> 1.	9	class cannot be inherited.
# ### 4m#	10	This class specifies that the incoming data will be in the format
70	11	"name=value&name=value", encoded from an
	12	System.Web.UI.HtmlControls.HtmlForm, whether the value of the HtmlForm's
	13	System.Web.UI.HtmlControls.HtmlForm.Method property is POST or GET.
	14	HttpUrlEncodedBinding
	15	Example Syntax:
	16	ToString
	17	
	18	[C#] public HttpUrlEncodedBinding();
	19	[C++] public: HttpUrlEncodedBinding();
	20	[VB] Public Sub New()
	21	[JScript] public function HttpUrlEncodedBinding();
	22	Handled
	23	Parent
	24	Required
	25	HttpUrlReplacementBinding class (System.Web.Services.Description)

Description

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Represents an extensibility element added to an System. Web. Services. Description. InputBinding within a Web Service. It specifies the format for data transmission through HTTP. This class cannot be inherited.

This class allows users to specify a custom format for data transmission, rather than limiting them to the standard format, "name=value&name=value", used in HTTP query strings. For example, a developer may decide to use /name,value/name,value or some other custom format instead.

HttpUrlReplacementBinding

Example Syntax:

**ToString** 

[C#] public HttpUrlReplacementBinding();

[C++] public: HttpUrlReplacementBinding();

[VB] Public Sub New()

[JScript] public function HttpUrlReplacementBinding();

Handled

Parent

Required

Import class (System.Web.Services.Description)

**ToString** 

Her Many Course of the Course

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Associates an XML namespace with a document location. This class cannot be inherited.

Many of the tags in the WSDL schema can contain the tag. It allows the separation of different elements of the Web Service into different documents, which can then be imported as needed.

**Import** 

Example Syntax:

**ToString** 

[C#] public Import();

[C++] public: Import();

[VB] Public Sub New()

[JScript] public function Import();

Documentation

Location

**ToString** 

Description

Gets or sets a reference to the XML Location attribute of the

System.Web.Services.Description.Import instance.

Namespace

1	ToString
2	
3	[C#] public string Namespace {get; set;}
4	[C++] public:property String* get_Namespace();public:property void
5	set_Namespace(String*);
6	[VB] Public Property Namespace As String
7	[JScript] public function get Namespace(): String; public function set
8	Namespace(String);
9	
10	Description
11	Gets or sets a reference to the XML Namespace of the
12	System.Web.Services.Description.Import instance.
13	ServiceDescription
14	ToString
15	
16	[C#] public ServiceDescription ServiceDescription {get;}
17	[C++] public:property ServiceDescription* get_ServiceDescription();
18	[VB] Public ReadOnly Property ServiceDescription As ServiceDescription
19	[JScript] public function get ServiceDescription(): ServiceDescription;
20	
21	Description
22	Gets a reference to the
23	System.Web.Services.Description.ServiceDescription of which the Import
24	instance is a member.
25	ImportCollection class (System.Web.Services.Description)

ToString

2

3

5

6

7

8

10

11

13

14

15 16

17

18

19

20

21

23

24

Description

Provides a collection of **System.Web.Services.Description.Import** instances representing documents to be imported into the Web Service. This class cannot be inherited.

Count

InnerList

Item

**ToString** 

Description

Gets or sets the value of an **System.Web.Services.Description.Import** at the specified zero-based index. The zero-based index of the **Import** whose value is modified or returned.

List

Table

Add

[C#] public int Add(Import import);

[C++] public: int Add(Import\* import);

[VB] Public Function Add(ByVal import As Import) As Integer

[JScript] public function Add(import : Import) : int;

Description

	11								
	4	the end of the System. Web. Services. Description. Import Collection.							
	5	Return Value: Returns the index where import has been added. The							
	6	System.Web.Services.Description.Import to be added to the collection.							
	7	Contains							
	8								
Paris Train	9	[C#] public bool Contains(Import import);							
	10	[C++] public: bool Contains(Import* import);							
	11	[VB] Public Function Contains(ByVal import As Import) As Boolean							
	12	[JScript] public function Contains(import : Import) : Boolean;							
	13								
	14	Description							
	15	Gets a value indicating whether the specified							
	16	System.Web.Services.Description.Import instance is a member of the							
	17	System.Web.Services.Description.ImportCollection .							
	18	Return Value: true if import is a member of the							
	19	System.Web.Services.Description.ImportCollection; otherwise, false. An							
	20	System.Web.Services.Description.Import object.							
	21	СоруТо							
	- 11	j							

[C#] public void CopyTo(Import[] array, int index);

[C++] public: void CopyTo(Import\* array[], int index);

Adds the specified System. Web. Services. Description. Import instance to

[VB] Public Sub CopyTo(ByVal array() As Import, ByVal index As Integer)

[JScript] public function CopyTo(array : Import[], index : int);

Description

2

3

6

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Copies the entire ImportCollection to a compatible one-dimensional array of type System.Web.Services.Description.Import, starting at the specified zero-based index of the target array. The destination array, of type System.Web.Services.Description.Import. The zero-based index at which to start placing the copied collection.

IndexOf

[C#] public int IndexOf(Import import);

[C++] public: int IndexOf(Import\* import);

[VB] Public Function IndexOf(ByVal import As Import) As Integer

[JScript] public function IndexOf(import : Import) : int;

Description

Searches for the specified **System.Web.Services.Description.Import** and returns the zero-based index of the first occurrence within the collection.

Return Value: Returns a 32-bit signed integer. An

System. Web. Services. Description. Import object.

Insert

[C#] public void Insert(int index, Import import);

[C++] public: void Insert(int index, Import\* import);

[VB] Public Sub Insert(ByVal index As Integer, ByVal import As Import)

[JScript] public function Insert(index: int, import: Import); 2 Description 3 Adds the specified System. Web. Services. Description. Import instance to 4 the ImportCollection at the specified index. If the number of items in the collection already equals the collection's 6 capacity, the capacity is doubled by automatically reallocating the internal array 7 before the new element is inserted. The zero-based index at which to insert import. The **System.Web.Services.Description.Import** to be added to the collection. Remove 10 11 [C#] public void Remove(Import import); [C++] public: void Remove(Import\* import); 13 [VB] Public Sub Remove(ByVal import As Import) [JScript] public function Remove(import : Import); 15 16 Description 17 Removes the first occurrence of the specified 18 System. Web. Services. Description. Import from the Import Collection. 19 This method performs a linear search; therefore, the average execution time 20 is proportional to System. Web. Services. Description. Import Collection. Count. 21 An System. Web. Services. Description. Import object. 22 SetParent 23 24

[C#] protected override void SetParent(object value, object parent);

1	[C++] protected: void SetParent(Object* value, Object* parent);
2	[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
3	Object)
4	[JScript] protected override function SetParent(value : Object, parent : Object);
5	
6	Description
7	Sets the parent System. Web. Services. Description. Service Description of
8	the specified System. Web. Services. Description. Import. An object, of type
9	System.Web.Services.Description.Import, within the collection. The object, of
10	type System. Web. Services. Description. Service Description, to set as the parent.
11	InputBinding class (System.Web.Services.Description)
12	ToString
13	
14	
15	Description
16	Provides a set of specifications for data formats and protocols used by the
17	Web Service for input messages. This class cannot be inherited.
18	InputBinding
19	Example Syntax:
20	ToString
21	
22	[C#] public InputBinding();
23	[C++] public: InputBinding();
24	[VB] Public Sub New()
25	[JScript] public function InputBinding();

1	Documentation
2	Extensions
3	ToString
4	
5	
6	Description
7	Gets the
8	System. We b. Services. Description. Service Description Format Extension Collect
9	ion associated with the InputBinding instance.
10	Name
11	OperationBinding
12	Message class (System.Web.Services.Description)
13	ToString
14	
15	
16	Description
17	Defines in the abstract the content, either document-oriented or procedure-
18	oriented, of data passed by the Web Service. This class cannot be inherited.
19	Message
20	Example Syntax:
21	ToString
22	
23	[C#] public Message();
24	[C++] public: Message();
25	

	1	[VB] Public Sub New()
	2	[JScript] public function Message();
	3	Documentation
	4	Name
	5	ToString
	6	
	7	
	8	Description
w <sub>i</sub>	9	Gets or sets the name of the Message instance.
"看着"看着"	10	Parts
These thinks that there the target than the target	11	ToString
	12	
7,777	13	[C#] public MessagePartCollection Parts {get;}
	14	[C++] public:property MessagePartCollection* get_Parts();
	15	[VB] Public ReadOnly Property Parts As MessagePartCollection
	16	[JScript] public function get Parts(): MessagePartCollection;
	17	
	18	Description
	19	Gets the collection of all the
	20	System.Web.Services.Description.MessagePart objects contained in the
	21	Message instance.
	22	ServiceDescription
	23	ToString
	24	
	25	[C#] public ServiceDescription ServiceDescription {get;}

[C++] public: property ServiceDescription\* get ServiceDescription(); [VB] Public ReadOnly Property ServiceDescription As ServiceDescription 2 [JScript] public function get ServiceDescription(): ServiceDescription; 3 Description 5 Gets the System. Web. Services. Description. Service Description of which 6 the Message instance is a member. 7 FindPartByName 8 9 [C#] public MessagePart FindPartByName(string partName); 10 [C++] public: MessagePart\* FindPartByName(String\* partName); 11 [VB] Public Function FindPartByName(ByVal partName As String) As 12 MessagePart 13 [JScript] public function FindPartByName(partName : String) : MessagePart; 14 15 Description 16 Searches the System. Web. Services. Description. Message Part Collection 17 returned by the System. Web. Services. Description. Message. Parts property, and 18 returns the named System. Web. Services. Description. Message Part. The string 19 naming the System. Web. Services. Description. Message Part to be returned. 20 FindPartsByName 21 22 [C#] public MessagePart[] FindPartsByName(string[] partNames); 23 [C++] public: MessagePart\* FindPartsByName(String\* partNames gc[]) []; 24 [VB] Public Function FindPartsByName(ByVal partNames() As String) As

MessagePart()
[JScript] public function FindPartsByName(partNames : String[]) : MessagePart[];
Description
Searches the System. Web. Services. Description. Message Part Collection
returned by the System. Web. Services. Description. Message. Parts property and
returns an array of type System. Web. Services. Description. Message Part
containing the named instances.
Return Value: An array of type System. Web. Services. Description. Message Part.
An array of names of the System. Web. Services. Description. Message Part
instances to be returned.
MessageBinding class (System.Web.Services.Description)
ToString
Description
Describes how abstract content is mapped into a concrete format. This
abstract class forms the base class for the following classes:
System.Web.Services.Description.FaultBindingSystem.Web.Services.Descript
ion.InputBindingSystem.Web.Services.Description.OutputBinding
MessageBinding
Example Syntax:
ToString
[C#] protected MessageBinding();

```
[C++] protected: MessageBinding();
    [VB] Protected Sub New()
    [JScript] protected function MessageBinding();
3
           Documentation
           Extensions
5
           ToString
8
    Description
           Gets the
10
    System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect
11
    ion associated with the MessageBinding instance.
12
           This property is intended to be overridden to provide the format extensions
13
    for Input, Output and Fault messages.
14
           Name
15
           ToString
16
17
    [C#] public string Name {get; set;}
18
    [C++] public: __property String* get_Name();public: __property void
19
    set Name(String*);
20
    [VB] Public Property Name As String
21
    [JScript] public function get Name(): String; public function set Name(String);
22
23
    Description
24
25
```

	1	Gets or sets the name of the
	2	System.Web.Services.Description.MessageBinding.
	3	OperationBinding
	4	ToString
	5	
	6	[C#] public OperationBinding OperationBinding {get;}
	7	[C++] public:property OperationBinding* get_OperationBinding();
	8	[VB] Public ReadOnly Property OperationBinding As OperationBinding
	9	[JScript] public function get OperationBinding(): OperationBinding;
51 St. 10	10	
	11	Description
nor one had been had but been been	12	Gets the System.Web.Services.Description.OperationBinding of which
of the state of th	13	the MessageBinding instance is a member.
- Addison	14	MessageCollection class (System.Web.Services.Description)
:	15	ToString
	16	
	17	
	18	Description
	19	Represents a collection of System.Web.Services.Description.Message
	20	objects. This class cannot be inherited.
	21	Count
	22	InnerList
	23	Item
	24	ToString

## System.Web.Services.Description.Message

2

.

5

٤

7

8

9

11

12

13

14

15

16

17 18

19

20

21

23

24

25

Description

Gets or sets the value of a **System.Web.Services.Description.Message** at the specified zero-based index. The zero-based index of the **Message** whose value is modified or returned.

Item

**ToString** 

[C#] public Message this[string name] {get;}

[C++] public: \_\_property Message\* get\_Item(String\* name);

[VB] Public Default ReadOnly Property Item(ByVal name As String) As Message
[JScript] returnValue = MessageCollectionObject.Item(name);

Description

Gets the Message by its

**System.Web.Services.Description.Message.Name** property. The string value representing the name of the **Message** returned.

List

Table

Add

[C#] public int Add(Message message);

[C++] public: int Add(Message\* message);

[VB] Public Function Add(ByVal message As Message) As Integer

1	[JScript] public function Add(message : Message) : int;
2	
3	Description
4	Adds the specified System. Web. Services. Description. Message instance to
5	the end of the MessageCollection.
6	Return Value: Returns the zero-based index where message has been added. The
7	System.Web.Services.Description.Message to be added to the
8	MessageCollection .
9	Contains
10	
11	[C#] public bool Contains(Message message);
12	[C++] public: bool Contains(Message* message);
13	[VB] Public Function Contains(ByVal message As Message) As Boolean
14	[JScript] public function Contains(message : Message) : Boolean;
15	
16	Description
17	Gets a value indicating whether the specified
18	System.Web.Services.Description.Message instance is a member of the
19	MessageCollection .
20	Return Value: true if message is a member of the MessageCollection; otherwise,
21	false . A System.Web.Services.Description.Message object.
22	СоруТо
23	
24	[C#] public void CopyTo(Message[] array, int index);
25	[C++] public: void CopyTo(Message* array[], int index);

19

20

21

22

23

24

[VB] Public Sub CopyTo(ByVal array() As Message, ByVal index As Integer)
[JScript] public function CopyTo(array : Message[], index : int);

## Description

2

3

5

Copies the entire MessageCollection to a compatible one-dimensional array of type System.Web.Services.Description.Message, starting at the specified zero-based index of the target array. The array, of type System.Web.Services.Description.Message, serving as the destination of the copy action. The zero-based index at which to start placing the copied collection.

GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String [JScript] protected override function GetKey(value : Object) : String;

#### Description

Supplies the name of the **System.Web.Services.Description.Message** associated with the value passed by reference. A

System. Web. Services. Description. Message instance for which to return the name.

IndexOf

[C#] public int IndexOf(Message message);

[C++] public: int IndexOf(Message\* message);

Harm the state of the state of

11

12

13

15

16

17

18

19

20

21

22

24

[VB] Public Function IndexOf(ByVal message As Message) As Integer [JScript] public function IndexOf(message : Message) : int;

#### Description

3

5

Searches for the specified **System.Web.Services.Description.Message** instance and returns the zero-based index of the first occurrence within the collection.

Return Value: Returns a 32-bit signed integer. A

System.Web.Services.Description.Message object for which to search the MessageCollection.

Insert

[C#] public void Insert(int index, Message message);

[C++] public: void Insert(int index, Message\* message);

[VB] Public Sub Insert(ByVal index As Integer, ByVal message As Message)

[JScript] public function Insert(index : int, message : Message);

#### Description

Adds the specified **System.Web.Services.Description.Message** instance to the **MessageCollection** at the specified index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert *message*. The **System.Web.Services.Description.Message** instance to add to the collection.

1	Remove
2	
3	[C#] public void Remove(Message message);
4	[C++] public: void Remove(Message* message);
5	[VB] Public Sub Remove(ByVal message As Message)
6	[JScript] public function Remove(message : Message);
7	
8	Description
9	Removes the first occurrence of the specified
10	System.Web.Services.Description.Message instance from the
11	MessageCollection .
12	This method performs a linear search; therefore, the average execution time
13	is proportional to System. Web. Services. Description. Message Collection. Count.
14	A System.Web.Services.Description.Message instance for which to search the
15	collection.
16	SetParent
17	
18	[C#] protected override void SetParent(object value, object parent);
19	[C++] protected: void SetParent(Object* value, Object* parent);
20	[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
21	Object)
22	[JScript] protected override function SetParent(value : Object, parent : Object);
23	
24	Description
25	

Sets the parent System.Web.Services.Description.ServiceDescription of a member of the MessageCollection . An object, of type System.Web.Services.Description.Message, within the collection. The object, of type System.Web.Services.Description.ServiceDescription, to be set as the parent.

MessagePart class (System.Web.Services.Description)

**ToString** 

# Description

Allows messages to be broken up into their logical units, with specific abstract information for each part. This class cannot be inherited.

MessagePart

Example Syntax:

**ToString** 

[C#] public MessagePart();

[C++] public: MessagePart();

[VB] Public Sub New()

[JScript] public function MessagePart();

Documentation

Element

**ToString** 

```
Description
       3
       5
       6
       7
       8
11
      12
      13
In the first first first first
      14
      15
             Description
      16
      17
      18
                       Name
      19
      20
      21
      22
```

```
Gets or sets the XML element name of the
    System.Web.Services.Description.MessagePart instance.
          This property corresponds to the attribute of the tag for which the
    MessagePart class serves as a wrapper.
          Message
          ToString
    [C#] public Message Message {get;}
    [C++] public: property Message* get Message();
    [VB] Public ReadOnly Property Message As Message
    [JScript] public function get Message(): Message;
          Gets the System. Web. Services. Description. Message of which the
    MessagePart instance is a member.
          ToString
    [C#] public string Name {get; set;}
    [C++] public: property String* get Name();public: property void
23
    set Name(String*);
24
    [VB] Public Property Name As String
```

[JScript] public function get Name(): String; public function set Name(String);

|| Description

2

3

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the name of the

#### System. Web. Services. Description. Message Part instance.

Returns an empty string ("") if the property value has not been assigned.

This property corresponds to the attribute of the tag for which the MessagePart class serves as a wrapper.

Type

**ToString** 

[C#] public XmlQualifiedName Type {get; set;}

[C++] public: \_\_property XmlQualifiedName\* get\_Type();public: \_\_property void set Type(XmlQualifiedName\*);

[VB] Public Property Type As XmlQualifiedName

[JScript] public function get Type() : XmlQualifiedName;public function set Type(XmlQualifiedName);

Description

Gets or sets the XML datatype of the

#### System.Web.Services.Description.MessagePart instance.

This property corresponds to the attribute of the tag for which the MessagePart class serves as a wrapper. In general it refers to an XSD datatype, but can also be extended as long as the XML namespace used is different from that of WSDL.

1	MessagePartCollection class (System.Web.Services.Description)
2	ToString
3	
4	
5	Description
6	Represents a collection of
7	System.Web.Services.Description.MessagePart objects. This class cannot be
8	inherited.
9	Count
10	InnerList
11	Item
12	ToString
13	System.Web.Services.Description.MessagePart
14	
15	Description
16	Gets or sets the value of a
17	System.Web.Services.Description.MessagePart at the specified zero-based
18	index. The zero-based index of the <b>MessagePart</b> whose value is modified or
19	returned.
20	Item
21	ToString
22	
23	[C#] public MessagePart this[string name] {get;}
24	[C++] public:property MessagePart* get_Item(String* name);
25	[VB] Public Default ReadOnly Property Item(ByVal name As String) As

1	MessagePart
2	[JScript] returnValue = MessagePartCollectionObject.Item(name);
3	
4	Description
5	Gets the MessagePart by its Name property. The string value representing
6	the name of the System. Web. Services. Description. Message Part returned.
7	List
8	Table
9	Add
10	
11	[C#] public int Add(MessagePart messagePart);
12	[C++] public: int Add(MessagePart* messagePart);
13	[VB] Public Function Add(ByVal messagePart As MessagePart) As Integer
14	[JScript] public function Add(messagePart : MessagePart) : int;
15	
16	Description
17	Adds the specified System. Web. Services. Description. Message Part to the
18	$end\ of\ the\ {\bf System. Web. Services. Description. Message Part Collection}\ .$
19	Return Value: Returns the index where messagePart has been added. The
20	System.Web.Services.Description.MessagePart to be added to the collection.
21	Contains
22	
23	[C#] public bool Contains(MessagePart messagePart);
24	[C++] public: bool Contains(MessagePart* messagePart);
25	[VB] Public Function Contains(ByVal messagePart As MessagePart) As Boolean

[JScript] public function Contains(messagePart : MessagePart) : Boolean; 1 2 Description 3 Gets a value indicating whether the specified 4 System. Web. Services. Description. Message Part is a member of the 5 MessagePartCollection. 6 Return Value: true if messagePart is a member of the MessagePartCollection; 7 otherwise, false. A System. Web. Services. Description. Message Part object. 8 CopyTo 9 10 [C#] public void CopyTo(MessagePart[] array, int index); 11 [C++] public: void CopyTo(MessagePart\* array[], int index); 12 [VB] Public Sub CopyTo(ByVal array() As MessagePart, ByVal index As Integer) 13 [JScript] public function CopyTo(array: MessagePart[], index: int); 14 15 Description 16 Copies the entire MessagePartCollection to a compatible one-dimensional 17 array of type System. Web. Services. Description. Message Part, starting at the 18 specified zero-based index of the target array. An array of type 19 System. Web. Services. Description. Message Part serving as the destination of the 20 copy action. The zero-based index at which to start placing the copied collection. 21 GetKey 22 23 [C#] protected override string GetKey(object value); 24 [C++] protected: String\* GetKey(Object\* value);

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String [JScript] protected override function GetKey(value : Object) : String; Description Returns the name of the System. Web. Services. Description. Message Part associated with the value passed by reference. An object for which to return the name. IndexOf [C#] public int IndexOf(MessagePart messagePart); [C++] public: int IndexOf(MessagePart\* messagePart); [VB] Public Function IndexOf(ByVal messagePart As MessagePart) As Integer [JScript] public function IndexOf(messagePart : MessagePart) : int; Description Searches for the specified System.Web.Services.Description.MessagePart and returns the zero-based index of the first occurrence within the collection. Return Value: Returns a 32-bit signed integer. A System.Web.Services.Description.MessagePart object. Insert [C#] public void Insert(int index, MessagePart messagePart); [C++] public: void Insert(int index, MessagePart\* messagePart); [VB] Public Sub Insert(ByVal index As Integer, ByVal messagePart As

	2
	3
	4
	5
	6
	7
	8
	9
and Am	10
	11
f V	12
this this	13
u de la composition della comp	14
	15
ing gray	16
-	17
	18
	19
	20
	21
	22
	23

MessagePart)	essagePart	:)
--------------	------------	----

[JScript] public function Insert(index: int, messagePart: MessagePart);

Description

Adds the specified **System.Web.Services.Description.MessagePart** to the **MessagePartCollection** at the specified index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert *messagePart*. The **System.Web.Services.Description.MessagePart** to be added to the collection.

Remove

[C#] public void Remove(MessagePart messagePart);

[C++] public: void Remove(MessagePart\* messagePart);

[VB] Public Sub Remove(ByVal messagePart As MessagePart)

[JScript] public function Remove(messagePart : MessagePart);

Description

Removes the first occurrence of the specified

System.Web.Services.Description.MessagePart from the

MessagePartCollection .

This method performs a linear search; therefore, the average execution time is proportional to

24

25

System.Web.Services.Description.MessagePartCollection.Count . A
System.Web.Services.Description.MessagePart object.

SetParent

[C#] protected override void SetParent(object value, object parent);

[C++] protected: void SetParent(Object\* value, Object\* parent);

[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As Object)

[JScript] protected override function SetParent(value : Object, parent : Object);

#### Description

Sets the parent System.Web.Services.Description.Message of a member of System.Web.Services.Description.MessagePartCollection . An object, of type System.Web.Services.Description.MessagePart, within the collection. The object, of type System.Web.Services.Description.Message, to set as the parent.

MimeContentBinding class (System.Web.Services.Description)
ToString

#### Description

Represents an extensibility element added to an 
System.Web.Services.Description.InputBinding or an 
System.Web.Services.Description.OutputBinding within a Web Service, 
specifying the MIME format for the body of the HTTP transmission. This class cannot be inherited.

MS1-863US.APP

	1	ToString
	2	
	3	[C#] public const string Namespace;
	4	[C++] public: const String* Namespace;
	5	[VB] Public Const Namespace As String
	6	[JScript] public var Namespace : String;
	7	
	8	Description
	9	Specifies the URI for the XML namespace of the MimeContentBinding
	10	class. This field is constant.
Sant And Mind Store And Sant Start Start	11	MimeContentBinding
H	12	Example Syntax:
	13	ToString
Н	14	
	15	[C#] public MimeContentBinding();
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	16	[C++] public: MimeContentBinding();
ļ. <b>ā.</b>	17	[VB] Public Sub New()
	18	[JScript] public function MimeContentBinding();
	19	Handled
	20	Parent
	21	Part
	22	ToString
	23	

24 Description

Gets or sets the name of the

System.Web.Services.Description.MessagePart to which the

MimeContentBinding applies.

Required

Type

**ToString** 

Description

Gets or sets a value indicating the format of the body of the HTTP transmission.

The WSDL specification defines bindings for several MIME types, including text/xml, multipart/related, and application/x-www-form-urlencoded, although any MIME type may be used. The wildcard character (\*) may also be used. For example, the string "text/\*" would represent all text types. If the value of this property is not set, the **MimeContentBinding** specifies all MIME types.

MimeMultipartRelatedBinding class (System.Web.Services.Description)
ToString

Description

Represents an extensibility element added to an 
System.Web.Services.Description.InputBinding or an 
System.Web.Services.Description.OutputBinding, specifying the individual

	1	MIME formats for each System. Web. Services. Description. Message Part within
	2	the HTTP transmission. This class cannot be inherited.
	3	MimeMultipartRelatedBinding
	4	Example Syntax:
	5	ToString
	6	
	7	[C#] public MimeMultipartRelatedBinding();
	8	[C++] public: MimeMultipartRelatedBinding();
	9	[VB] Public Sub New()
: 5 <u>1</u>	10	[JScript] public function MimeMultipartRelatedBinding();
\$? 4? 4 4 4 4 4 4.	11	Handled
	12	Parent
#"# #"#	13	Parts
	14	ToString
Br. M. Hang	15	
#"# #"#	16	
zi.	17	Description
	18	Gets the collection of extensibility elements added to the
	19	MimeMultipartRelatedBinding, which specify the MIME format for the
	20	corresponding System. Web. Services. Description. Message Part instances.
	21	Required
	22	MimePart class (System.Web.Services.Description)
	23	ToString
	24	
	25	

	1	
	2	
	3	Description
	4	Represents an extensibility element added to a
	5	${\bf System. Web. Services. Description. Mime Multipart Related Binding}\ ,\ specifying$
	6	the concrete MIME type for the System. Web. Services. Description. Message Part
	7	to which the MimePart applies. This class cannot be inherited.
մոսի մուն կուն կուս չում մուն կուն հուն	8	MimePart
	9	Example Syntax:
	10	ToString
	11	
	12	[C#] public MimePart();
	13	[C++] public: MimePart();
	14	[VB] Public Sub New()
2000	15	[JScript] public function MimePart();
11. 1115. 11114.	16	Extensions
	17	ToString
	18	
	19	[C#] public ServiceDescriptionFormatExtensionCollection Extensions {get;}
	20	[C++] public:property ServiceDescriptionFormatExtensionCollection*
	21	get_Extensions();
	22	[VB] Public ReadOnly Property Extensions As
	23	ServiceDescriptionFormatExtensionCollection
	24	[JScript] public function get Extensions():
	25	ServiceDescriptionFormatExtensionCollection;

22

23

24

25

# Description

3

Gets the collection of bindings for the

System. Web. Services. Description. MimeMultipartRelatedBinding of which the

MimePart is a member.

This collection must have at least one member. If it has more than one member, then the members are alternative MIME formats for a corresponding System.Web.Services.Description.MessagePart.

Handled

Parent

Required

MimePartCollection class (System.Web.Services.Description)

**ToString** 

## Description

Represents a collection of **System.Web.Services.Description.MimePart** objects. This class cannot be inherited.

MimePartCollection

Example Syntax:

**ToString** 

[C#] public MimePartCollection();

[C++] public: MimePartCollection();

[VB] Public Sub New() [JScript] public function MimePartCollection(); 2 Count 3 InnerList Item 5 **ToString** 6 7 8 Description 9 Gets or sets the value of a System. Web. Services. Description. MimePart at 10 the specified zero-based index. The zero-based index of the MimePart whose 11 value is modified or returned. 12 List 13 Add 14 15 [C#] public int Add(MimePart mimePart); 16 [C++] public: int Add(MimePart\* mimePart); 17 [VB] Public Function Add(ByVal mimePart As MimePart) As Integer 18 [JScript] public function Add(mimePart : MimePart) : int; 19 20 Description 21 Adds the specified System. Web. Services. Description. MimePart instance 22 to the end of the MimePartCollection . 23 Return Value: Returns the zero-based index where mimePart has been added. The

System. Web. Services. Description. MimePart to be added to the collection.

$\sim$	
('An	tains
COH	tams

1

3

\_

5

6

7

9

10

11

13 14

15 16

17

18 19

20

2223

24

25

[C#] public bool Contains(MimePart mimePart);

[C++] public: bool Contains(MimePart\* mimePart);

[VB] Public Function Contains(ByVal mimePart As MimePart) As Boolean

[JScript] public function Contains(mimePart : MimePart) : Boolean;

Description

Gets a value indicating whether the specified

**System.Web.Services.Description.MimePart** instance is a member of the **MimePartCollection** .

Return Value: true if mimePart is a member of the MimePartCollection; otherwise, false. A System.Web.Services.Description.MimePart instance.

CopyTo

[C#] public void CopyTo(MimePart[] array, int index);

[C++] public: void CopyTo(MimePart\* array[], int index);

[VB] Public Sub CopyTo(ByVal array() As MimePart, ByVal index As Integer)

[JScript] public function CopyTo(array : MimePart[], index : int);

Description

Copies the entire MimePartCollection to a compatible one-dimensional array of type System.Web.Services.Description.MimePart, starting at the specified zero-based index of the target array. An array of type

	1	System.Web.Services.Description.MimePart serving as the destination of the
	2	copy action. The zero-based index at which to start placing the copied collection.
	3	IndexOf
	4	
	5	[C#] public int IndexOf(MimePart mimePart);
	6	[C++] public: int IndexOf(MimePart* mimePart);
	7	[VB] Public Function IndexOf(ByVal mimePart As MimePart) As Integer
	8	[JScript] public function IndexOf(mimePart : MimePart) : int;
	9	
	10	Description
	11	Searches for the specified System. Web. Services. Description. MimePart
	12	instance and returns the zero-based index of the first occurrence within the
had had had ham had had had had	13	collection.
	14	Return Value: Returns a 32-bit signed integer. A
Mr. Berlin Mer. Bron. Start	15	System.Web.Services.Description.MimePart object for which to search the
	16	MimePartCollection.
Ž	17	Insert
	18	
	19	[C#] public void Insert(int index, MimePart mimePart);
	20	[C++] public: void Insert(int index, MimePart* mimePart);
	21	[VB] Public Sub Insert(ByVal index As Integer, ByVal mimePart As MimePart)
	22	[JScript] public function Insert(index : int, mimePart : MimePart);
	23	
	24	Description
	25	

3

6

7

8

9

10

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Adds the specified **System.Web.Services.Description.MimePart** instance to the **MimePartCollection** at the specified index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert *mimePart*. The **System.Web.Services.Description.MimePart** to add to the collection.

Remove

[C#] public void Remove(MimePart mimePart);

[C++] public: void Remove(MimePart\* mimePart);

[VB] Public Sub Remove(ByVal mimePart As MimePart)

[JScript] public function Remove(mimePart : MimePart);

Description

Removes the first occurrence of the specified

 ${\bf System. Web. Services. Description. Mime Part\ from\ the\ Mime Part\ Collection\ .}$ 

This method performs a linear search; therefore, the average execution time is proportional to

System.Web.Services.Description.MimePartCollection.Count . A System.Web.Services.Description.MimePart instance for which to search the collection.

MimeTextBinding class (System.Web.Services.Description)

**ToString** 

	- 11	
	1	
	2	
	3	Description
	4	Represents an extensibility element added to an
	5	System.Web.Services.Description.InputBinding, an
	6	System.Web.Services.Description.OutputBinding or a
	7	System.Web.Services.Description.MimePart, specifying the text patterns for
	8	which to search the HTTP transmission. This class cannot be inherited.
	9	ToString
inii.	10	
նում։ Կում։ Կում։ Աստ Ուսմ։ Արդի կում։ Կում։	11	[C#] public const string Namespace;
	12	[C++] public: const String* Namespace;
and Think The	13	[VB] Public Const Namespace As String
	14	[JScript] public var Namespace : String;
	15	
ing, in	16	Description
	17	Specifies the URI for the XML namespace of the MimeTextBinding class.
	18	This field is constant.
	19	MimeTextBinding
	20	Example Syntax:
	21	ToString
	22	
	23	[C#] public MimeTextBinding();
	24	[C++] public: MimeTextBinding();
	25	

[C++] public: MimeTextMatch();

results of the text search.

24

25

lee@haves pilc 509-324-9256 524 MS1-863US.APP

	1	The value of this property should be less than or equal to the value of the
	2	System.Web.Services.Description.MimeTextMatch.Repeats property. It is
	3	recommended that developers use the default value.
	4	IgnoreCase
	5	ToString
	6	
	7	[C#] public bool IgnoreCase {get; set;}
	8	[C++] public:property bool get_IgnoreCase();public:property void
	9	set_IgnoreCase(bool);
7	10	[VB] Public Property IgnoreCase As Boolean
	11	[JScript] public function get IgnoreCase(): Boolean; public function set
	12	IgnoreCase(Boolean);
	13	
	14	Description
	15	Gets or sets a value indicating whether the search should disregard the case
**************************************	16	of the text to be searched.
	17	Matches
	18	ToString
	19	
	20	[C#] public MimeTextMatchCollection Matches {get;}
	21	[C++] public:property MimeTextMatchCollection* get_Matches();
	22	[VB] Public ReadOnly Property Matches As MimeTextMatchCollection
	23	[JScript] public function get Matches(): MimeTextMatchCollection;
	24	
	25	Description

- 11	
1	
2	Name
3	ToString
4	
5	[C#] public string Name {get; set;}
6	[C++] public:property String* get_Name();public:property void
7	set_Name(String*);
8	[VB] Public Property Name As String
9	[JScript] public function get Name(): String; public function set Name(String);
10	
11	Description
12	Gets or sets the name of the MimeTextMatch instance.
13	Pattern
14	ToString
15	
16	[C#] public string Pattern {get; set;}
17	[C++] public:property String* get_Pattern();public:property void
18	set_Pattern(String*);
19	[VB] Public Property Pattern As String
20	[JScript] public function get Pattern(): String; public function set Pattern(String);
21	
22	Description
23	Gets or sets the text pattern for the search.
24	The value of this property can contain wildcard characters.
25	Repeats

1	ToString
2	
3	[C#] public int Repeats {get; set;}
4	[C++] public:property int get_Repeats();public:property void
5	set_Repeats(int);
6	[VB] Public Property Repeats As Integer
7	[JScript] public function get Repeats(): int;public function set Repeats(int);
8	
9	Description
10	Gets or sets a value indicating the number of times the search is to be
11	performed.
12	In a case where there is more than one possible match, a property value of 1
13	will return only the first match found.
14	RepeatsString
15	ToString
16	
17	[C#] public string RepeatsString {get; set;}
18	[C++] public:property String* get_RepeatsString();public:property void
19	set_RepeatsString(String*);
20	[VB] Public Property RepeatsString As String
21	[JScript] public function get RepeatsString(): String;public function set
22	RepeatsString(String);
23	
24	Description
25	

performed. 2 This property returns the same information as the 3 System.Web.Services.Description.MimeTextMatch.Repeats property, but as a string rather than a 32-bit signed integer. A value of "\*" corresponds to 5 System.Int32.MaxValue. Type 7 **ToString** 8 9 [C#] public string Type {get; set;} 10 [C++] public: \_\_property String\* get\_Type();public: \_\_property void 11 set Type(String\*); 12 [VB] Public Property Type As String 13 [JScript] public function get Type() : String; public function set Type(String); 14 15 Description 16 Gets or sets a value indicating the MIME format of the text to be searched. 17 This property returns an empty string ("") if its value has not been set. 18 MimeTextMatchCollection class (System.Web.Services.Description) 19 **ToString** 20 21 22 Description 23 24 25

Gets or sets a value indicating the number of times the search is to be

22

23

24

Provides a collection of 1 System.Web.Services.Description.MimeTextMatch instances. This class cannot be inherited. 3 MimeTextMatchCollection Example Syntax: **ToString** 6 7 [C#] public MimeTextMatchCollection(); [C++] public: MimeTextMatchCollection(); 9 [VB] Public Sub New() 10 [JScript] public function MimeTextMatchCollection(); 11 Count 12 InnerList 13 Item 14 **ToString** 15 16 17 Description 18 Gets or sets the value of a member of the MimeTextMatchCollection at 19 the specified zero-based index. The zero-based index of the MimeTextMatch 20

whose value is returned or modified.

List

Add

[C#] public int Add(MimeTextMatch match);

	1	[C++] public: int Add(MimeTextMatch* match);
	2	[VB] Public Function Add(ByVal match As MimeTextMatch) As Integer
	3	[JScript] public function Add(match : MimeTextMatch) : int;
	4	
	5	Description
	6	Adds the specified System.Web.Services.Description.MimeTextMatch
	7	instance to the end of the MimeTextMatchCollection.
	8	Return Value: Returns the zero-based index where match has been added. The
	9	System.Web.Services.Description.MimeTextMatch to be added to the
	10	collection.
inar tani tani tana tani tani tani tani	11	Contains
1111, MAGE - 11	12	
	13	[C#] public bool Contains(MimeTextMatch match);
	14	[C++] public: bool Contains(MimeTextMatch* match);
200000 000000	15	[VB] Public Function Contains(ByVal match As MimeTextMatch) As Boolean
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	16	[JScript] public function Contains(match : MimeTextMatch) : Boolean;
2	17	
	18	Description
	19	Gets a value indicating whether the specified
	20	System.Web.Services.Description.MimeTextMatch instance is a member of the
	21	MimeTextMatchCollection .
	22	Return Value: true if match is a member of the MimeTextMatchCollection;
	23	otherwise, false . A System.Web.Services.Description.MimeTextMatch
	24	instance.
	25	СоруТо

17

18

19

20

21

22

23

24

1 [C#] public void CopyTo(MimeTextMatch[] array, int index); 2 [C++] public: void CopyTo(MimeTextMatch\* array[], int index); 3 [VB] Public Sub CopyTo(ByVal array() As MimeTextMatch, ByVal index As Integer) 5 [JScript] public function CopyTo(array: MimeTextMatch[], index: int); 7 Description 8 Copies the entire MimeTextMatchCollection to a compatible one-9  $dimensional\ array\ of\ type\ {\bf System. Web. Services. Description. Mime Text Match}\ ,$ 10 starting at the specified zero-based index of the target array. An array of type System. Web. Services. Description. MimeTextMatch serving as the destination of 12 the copy action. The zero-based index at which to start placing the copied 13 collection. 14 IndexOf 15

[C#] public int IndexOf(MimeTextMatch match);

[C++] public: int IndexOf(MimeTextMatch\* match);

[VB] Public Function IndexOf(ByVal match As MimeTextMatch) As Integer

[JScript] public function IndexOf(match: MimeTextMatch): int;

Description

Searches for the specified

**System.Web.Services.Description.MimeTextMatch** instance and returns the zero-based index of the first occurrence within the collection.

Return Value: Returns a 32-bit signed integer. A System. Web. Services. Description. Mime Text Match object for which to search [C#] public void Insert(int index, MimeTextMatch match); [C++] public: void Insert(int index, MimeTextMatch\* match); [VB] Public Sub Insert(ByVal index As Integer, ByVal match As [JScript] public function Insert(index : int, match : MimeTextMatch); Adds the specified System. Web. Services. Description. MimeTextMatch instance to the **MimeTextMatchCollection** at the specified index. If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert match. The System. Web. Services. Description. MimeTextMatch to add to the collection. [C#] public void Remove(MimeTextMatch match); [C++] public: void Remove(MimeTextMatch\* match); [VB] Public Sub Remove(ByVal match As MimeTextMatch)

23

24

Description

5	MimeTextMatchCollection .
6	This method performs a linear search; therefore, the average execution time
7	is proportional to
8	System.Web.Services.Description.MimeTextMatchCollection.Count . A
9	System.Web.Services.Description.MimeTextMatch instance for which to search
10	the collection.
11	MimeXmlBinding class (System.Web.Services.Description)
12	ToString
13	
14	
15	Description
16	Represents an extensibility element added to a
17	System.Web.Services.Description.MimePart, an
18	System.Web.Services.Description.InputBinding or an
19	System.Web.Services.Description.OutputBinding. Specifies the schema for
20	XML messages that are not SOAP compliant. This class cannot be inherited.
21	MimeXmlBinding

Example Syntax:

[C#] public MimeXmlBinding();

**ToString** 

Removes the first occurrence of the specified

 ${\bf System. Web. Services. Description. Mime Text Match} \ instance \ from \ the$ 

```
[C++] public: MimeXmlBinding();
    [VB] Public Sub New()
2
    [JScript] public function MimeXmlBinding();
3
           Handled
           Parent
           Part
           ToString
9
    Description
10
           Gets or sets the name of the
11
    System. Web. Services. Description. Message Part to which the MimeXmlBinding
12
    applies.
13
           Required
14
           Operation class (System. Web. Services. Description)
15
           ToString
16
17
18
    Description
19
           Provides an abstract definition of an action supported by the Web Service.
20
    This class cannot be inherited.
21
           Exactly one instance of this class will be a member of the
22
    System.Web.Services.Description.Operation.Messages property of the parent
23
    System.Web.Services.Description.Operation.
           Operation
```

```
Example Syntax:
          ToString
2
3
   [C#] public Operation();
   [C++] public: Operation();
    [VB] Public Sub New()
    [JScript] public function Operation();
7
          Documentation
8
          Faults
9
           ToString
10
11
12
    Description
13
           Gets the collection of faults, or error messages, defined by the
14
    System. Web. Services. Description. Operation instance.
15
           Messages
16
           ToString
17
18
    [C#] public OperationMessageCollection Messages {get;}
19
    [C++] public: __property OperationMessageCollection* get_Messages();
20
     [VB] Public ReadOnly Property Messages As OperationMessageCollection
21
     [JScript] public function get Messages(): OperationMessageCollection;
22
23
     Description
24
25
```

2

3

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets the collection of **System.Web.Services.Description.Message** instances defined by the **System.Web.Services.Description.Operation** instance.

Because an Operation instance is associated with exactly one

System.Web.Services.Description.OperationInput and exactly one

System.Web.Services.Description.OperationOutput, only one instance of each can be a member of this collection, and thus the collection can have a maximum of two members.

Name

**ToString** 

```
[C#] public string Name {get; set;}
```

[C++] public: \_\_property String\* get\_Name();public: \_\_property void
set Name(String\*);

[VB] Public Property Name As String

[JScript] public function get Name(): String; public function set Name(String);

## Description

Gets or sets the name of the **System.Web.Services.Description.Operation** instance.

Returns an empty string ("") if the property value has not been set.

ParameterOrder

**ToString** 

[C#] public string[] ParameterOrder {get; set;}

[C++] public: \_\_property String\* get\_ParameterOrder();public: \_\_property void

1	set_ParameterOrder(String*gc[]);
2	[VB] Public Property ParameterOrder As String ()
3	[JScript] public function get ParameterOrder() : String[];public function set
4	ParameterOrder(String[]);
5	
6	Description
7	Gets or sets an array of elements contained in the
8	System.Web.Services.Description.Operation.ParameterOrderString.
9	Note that the elements of this array are identical to those of the
10	System.Web.Services.Description.Operation.ParameterOrderString, but
11	housed in an array rather than a space-delimited string.
12	ParameterOrderString
13	ToString
14	
15	[C#] public string ParameterOrderString {get; set;}
16	[C++] public:property String* get_ParameterOrderString();public:property
17	void set_ParameterOrderString(String*);
18	[VB] Public Property ParameterOrderString As String
19	[JScript] public function get ParameterOrderString(): String;public function set
20	ParameterOrderString(String);
21	
22	Description
23	An optional RPC-signature ordering specification for request/response or
24	solicit/response operations. The value is a list of names of
25	

space. 2 The named System. Web. Services. Description. Message Part instances 3 must adhere to the following rules: The order of the named System.Web.Services.Description.MessagePart instances reflects the order of the parameters in the RPC signature. PortType 7 **ToString** 8 9 [C#] public PortType PortType {get;} 10 [C++] public: property PortType\* get PortType(); 11 [VB] Public ReadOnly Property PortType As PortType 12 [JScript] public function get PortType() : PortType; 13 14 Description 15 Gets the System. Web. Services. Description. Port Type of which the 16 Operation is a member. IsBoundBy 18 19 [C#] public bool IsBoundBy(OperationBinding operationBinding); 20 [C++] public: bool IsBoundBy(OperationBinding\* operationBinding); 21 [VB] Public Function IsBoundBy(ByVal operationBinding As OperationBinding) As Boolean 23 [JScript] public function IsBoundBy(operationBinding: OperationBinding): 24 Boolean;

System. Web. Services. Description. Message Part instances separated by a single

Description

2

3

4

5

6

7

8

11

12

13

14

15

16

17

18

19

20

21

22

23

Gets a value indicating whether the specified

**System.Web.Services.Description.OperationBinding** is used in the **Operation** instance.

Return Value: true if the Operation instance uses operationBinding; otherwise,

false. An System.Web.Services.Description.OperationBinding instance to be checked to determine whether it is used by the Operation.

OperationBinding class (System.Web.Services.Description)

**ToString** 

Description

Provides specifications for protocols and data formats for the messages used in the action supported by the Web Service. This class cannot be inherited.

This class contains all the **System.Web.Services.Description.Binding** elements required for the actions supported by the Web Service.

**OperationBinding** 

Example Syntax:

**ToString** 

[C#] public OperationBinding();

[C++] public: OperationBinding();

[VB] Public Sub New()

[JScript] public function OperationBinding();

Binding **ToString** 2 3 [C#] public Binding Binding {get;} [C++] public: property Binding\* get Binding(); [VB] Public ReadOnly Property Binding As Binding [JScript] public function get Binding(): Binding; 8 Description The System. Web. Services. Description. Binding of which the 10 System. Web. Services. Description. Operation Binding instance is a member. 11 Documentation 12 **Extensions** 13 **ToString** 14 15 16 Description 17 Gets the 18 System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect 19 ion specific to the System. Web. Services. Description. Operation Binding 20 instance. 21 **Faults** 22 **ToString** 23 24

[C#] public FaultBindingCollection Faults {get;}

Name

25

[C++] public: \_\_property FaultBindingCollection\* get\_Faults(); [VB] Public ReadOnly Property Faults As FaultBindingCollection [JScript] public function get Faults(): FaultBindingCollection; Description 5 Gets the System. Web. Services. Description. Fault Binding Collection 6 associated with the System. Web. Services. Description. Operation Binding instance. Input **ToString** 11 [C#] public InputBinding Input {get; set;} 12 [C++] public: property InputBinding\* get Input();public: property void 13 set Input(InputBinding\*); 14 [VB] Public Property Input As InputBinding 15 [JScript] public function get Input(): InputBinding; public function set 16 Input(InputBinding); 17 18 Description 19 Gets or sets the System. Web. Services. Description. Input Binding 20 associated with the System. Web. Services. Description. Operation Binding 21 instance. 22 An OperationBinding will be associated with exactly one 23 System.Web.Services.Description.InputBinding. 24

1	ToString
2	
3	[C#] public string Name {get; set;}
4	[C++] public:property String* get_Name();public:property void
5	set_Name(String*);
6	[VB] Public Property Name As String
7	[JScript] public function get Name(): String; public function set Name(String);
8	
9	Description
10	Gets or sets the name of the OperationBinding instance.
11	In WSDL format, this property is realized as the name attribute of the tag
12	enclosed within the tags. Note that an empty string ("") is returned if this property
13	value has not been set.
14	Output
15	ToString
16	
17	[C#] public OutputBinding Output {get; set;}
18	[C++] public:property OutputBinding* get_Output();public:property void
19	set_Output(OutputBinding*);
20	[VB] Public Property Output As OutputBinding
21	[JScript] public function get Output(): OutputBinding;public function set
22	Output(OutputBinding);
23	
24	Description
25	

2

5

6

7

8

9

10

11

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets the **System.Web.Services.Description.OutputBinding** associated with the **System.Web.Services.Description.OperationBinding** instance.

 $\label{lem:constraint} \textbf{An OperationBinding} \ will \ be \ associated \ with \ exactly \ one \\ \textbf{System.Web.Services.Description.OutputBinding} \ .$ 

OperationBindingCollection class (System.Web.Services.Description)
ToString

## Description

Represents a collection of

System. Web. Services. Description. Operation Binding objects. This class cannot be inherited.

Count

InnerList

Item

**ToString** 

## Description

Gets or sets the value of a

**System.Web.Services.Description.OperationBinding** at the specified zero-based index. The zero-based index of the

**System.Web.Services.Description.OperationBinding** whose value is modified or returned.

22

23

24

25

Description

2

Gets a value indicating whether the specified

 ${\bf System. Web. Services. Description. Operation Binding \ is \ a \ member \ of \ the }$   ${\bf Operation Binding Collection} \ .$ 

Return Value: true if bindingOperation is a member of the

OperationBindingCollection; otherwise, false. An

System.Web.Services.Description.OperationBinding object.

CopyTo

[C#] public void CopyTo(OperationBinding[] array, int index);

[C++] public: void CopyTo(OperationBinding\* array[], int index);

[VB] Public Sub CopyTo(ByVal array() As OperationBinding, ByVal index As Integer)

[JScript] public function CopyTo(array: OperationBinding[], index: int);

Description

Copies the entire OperationBindingCollection to a compatible one-dimensional array of type System.Web.Services.Description.OperationBinding, starting at the specified zero-based index of the target array. An array of type System.Web.Services.Description.OperationBinding that serves as the destination for the copy action. The zero-based index at which to start placing the copied collection into the target array.

545

IndexOf

lee@haves onc 509-324-9256

Description

2

3

5

6

7

8

9

0

.2

.3

15

16

17

18

19

20

21

22

23

24

25

Gets a value indicating whether the specified

System. Web. Services. Description. Operation Binding is a member of the OperationBindingCollection .

Return Value: true if bindingOperation is a member of the

OperationBindingCollection; otherwise, false. An

System.Web.Services.Description.OperationBinding object.

CopyTo

[C#] public void CopyTo(OperationBinding[] array, int index);

[C++] public: void CopyTo(OperationBinding\* array[], int index);

[VB] Public Sub CopyTo(ByVal array() As OperationBinding, ByVal index As Integer)

[JScript] public function CopyTo(array : OperationBinding[], index : int);

Description

Copies the entire **OperationBindingCollection** to a compatible onedimensional array of type System. Web. Services. Description. Operation Binding , starting at the specified zero-based index of the target array. An array of type System.Web.Services.Description.OperationBinding that serves as the destination for the copy action. The zero-based index at which to start placing the copied collection into the target array.

IndexOf

1	
2	[C#] public int IndexOf(OperationBinding bindingOperation);
3	[C++] public: int IndexOf(OperationBinding* bindingOperation);
4	[VB] Public Function IndexOf(ByVal bindingOperation As OperationBinding) As
5	Integer
6	[JScript] public function IndexOf(bindingOperation : OperationBinding) : int;
7	
8	Description
9	Searches for the specified
10	System.Web.Services.Description.OperationBinding and returns the zero-based
11	index of the first occurrence within the OperationBindingCollection.
12	Return Value: Returns a 32-bit signed integer. An
13	System.Web.Services.Description.OperationBinding object.
14	Insert
15	
16	[C#] public void Insert(int index, OperationBinding bindingOperation);
17	[C++] public: void Insert(int index, OperationBinding* bindingOperation);
18	[VB] Public Sub Insert(ByVal index As Integer, ByVal bindingOperation As
19	OperationBinding)
20	[JScript] public function Insert(index : int, bindingOperation : OperationBinding);
21	
22	Description
23	Adds the specified System.Web.Services.Description.OperationBinding
24	instance to the OperationBindingCollection at the specified index.
25	

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert bindingOperation. The System.Web.Services.Description.OperationBinding to be added to the collection.

Remove

7

9

10

11

12

13

14

15

16

17

18

19

20

21

1

2

3

5

6

[C#] public void Remove(OperationBinding bindingOperation);

[C++] public: void Remove(OperationBinding\* bindingOperation);

[VB] Public Sub Remove(ByVal bindingOperation As OperationBinding)

[JScript] public function Remove(bindingOperation : OperationBinding);

Description

Removes the first occurrence of the specified

**System.Web.Services.Description.OperationBinding** from the **OperationBindingCollection** .

This method performs a linear search; therefore, the average execution time is proportional to

System.Web.Services.Description.OperationBindingCollection.Count . An System.Web.Services.Description.OperationBinding object.

SetParent

22 23

24

[C#] protected override void SetParent(object value, object parent);

[C++] protected: void SetParent(Object\* value, Object\* parent);

[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As

Object) [JScript] protected override function SetParent(value : Object, parent : Object); 2 3 Description Sets the parent System. Web. Services. Description. Binding of a member of 5 the OperationBindingCollection. An object, of type 6 System. Web. Services. Description. Operation Binding, within the collection. The 7 object, of type System. Web. Services. Description. Binding, to be set as the 8 parent. 9 OperationCollection class (System.Web.Services.Description) 10 **ToString** 11 12 13 Description 14 Represents a collection of System. Web. Services. Description. Operation 15 objects. This class cannot be inherited. 16 Count 17 InnerList 18 Item 19 **ToString** 20 21 22 Description 23 Gets or sets the value of a System. Web. Services. Description. Operation at 24 the specified zero-based index. The zero-based index of the 25

System. Web. Services. Description. Operation whose value is modified or returned. 2 List 3 Table Add 5 6 [C#] public int Add(Operation operation); 7 [C++] public: int Add(Operation\* operation); 8 [VB] Public Function Add(ByVal operation As Operation) As Integer 9 [JScript] public function Add(operation : Operation) : int; 10 11 Description 12 Adds the specified System. Web. Services. Description. Operation to the 13 end of the OperationCollection. 14 Return Value: Returns the zero-based index where operation has been added. The 15 System. Web. Services. Description. Operation to be added to the 16 OperationCollection. 17 Contains 18 19 [C#] public bool Contains(Operation operation); 20 [C++] public: bool Contains(Operation\* operation); 21 [VB] Public Function Contains(ByVal operation As Operation) As Boolean 22 [JScript] public function Contains(operation : Operation) : Boolean; 23 24 Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a value indicating whether the specified

System.Web.Services.Description.Operation is a member of the

OperationCollection.

Return Value: true if operation is a member of the OperationCollection;

otherwise, false. An System.Web.Services.Description.Operation object.

CopyTo

[C#] public void CopyTo(Operation[] array, int index);

[C++] public: void CopyTo(Operation\* array[], int index);

[VB] Public Sub CopyTo(ByVal array() As Operation, ByVal index As Integer)

[JScript] public function CopyTo(array: Operation[], index: int);

Description

Copies the entire **OperationCollection** to a compatible one-dimensional array of type **System.Web.Services.Description.Operation**, starting at the specified index of the target array. An array of type

**System.Web.Services.Description.Operation** serving as the destination for the copy action. The zero-based index at which to start placing the copied collection.

IndexOf

[C#] public int IndexOf(Operation operation);

[C++] public: int IndexOf(Operation\* operation);

[VB] Public Function IndexOf(ByVal operation As Operation) As Integer

[JScript] public function IndexOf(operation : Operation) : int;

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Searches for the specified **System.Web.Services.Description.Operation** and returns the zero-based index of the first occurrence within the **OperationCollection** .

Return Value: Returns a 32-bit signed integer. An

System.Web.Services.Description.Operation object.

Insert

[C#] public void Insert(int index, Operation operation);

[C++] public: void Insert(int index, Operation\* operation);

[VB] Public Sub Insert(ByVal index As Integer, ByVal operation As Operation)

[JScript] public function Insert(index : int, operation : Operation);

Description

Adds the specified **System.Web.Services.Description.Operation** instance to the **OperationCollection** at the specified index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert operation. The System.Web.Services.Description.Operation to be added to the OperationCollection.

Remove

[C#] public void Remove(Operation operation);

1	[C++] public: void Remove(Operation* operation);
2	[VB] Public Sub Remove(ByVal operation As Operation)
3	[JScript] public function Remove(operation : Operation);
4	
5	Description
6	Removes the first occurrence of the specified
7	System.Web.Services.Description.Operation from the OperationCollection .
8	This method performs a linear search; therefore, the average execution time
9	is proportional to
10	System. Web. Services. Description. Operation Collection. Count. The
11	System.Web.Services.Description.Operation object to be removed from the
12	Collection.
13	SetParent
14	
15	[C#] protected override void SetParent(object value, object parent);
16	[C++] protected: void SetParent(Object* value, Object* parent);
17	[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
18	Object)
19	[JScript] protected override function SetParent(value : Object, parent : Object);
20	
21	Description
22	Sets the parent <b>System.Web.Services.Description.PortType</b> of a member
23	of OperationCollection. An object, of type
24	System. Web. Services. Description. Operation, within the Operation Collection.
25	

1	The object, of type System. Web. Services. Description. Port Type, to set as the
2	parent.
3	OperationFault class (System.Web.Services.Description)
4	ToString
5	
6	
7	Description
8	Defines the specifications for error messages returned by the Web Service.
9	This class cannot be inherited.
10	Instances of this class will be members of the
11	System.Web.Services.Description.Operation.Faults property of the parent
12	Operation .
13	OperationFault
14	Example Syntax:
15	ToString
16	
17	[C#] public OperationFault();
18	[C++] public: OperationFault();
19	[VB] Public Sub New()
20	[JScript] public function OperationFault();
21	Documentation
22	Message
23	Name
24	Operation
25	OperationFaultCollection class (System.Web.Services.Description)

1	ToString
2	
3	
4	Description
5	Represents a collection of
6	System.Web.Services.Description.OperationFault objects. This class cannot be
7	inherited.
8	Count
9	InnerList
10	Item
11	ToString
12	System.Web.Services.Description.OperationFault
13	
14	Description
15	Gets or sets the value of a
16	System.Web.Services.Description.OperationFault at the specified zero-based
17	index. The zero-based index of the
18	System.Web.Services.Description.OperationFault whose value is modified or
19	returned.
20	Item
21	ToString
22	
23	[C#] public OperationFault this[string name] {get;}
24	[C++] public:property OperationFault* get_Item(String* name);
25	[VB] Public Default ReadOnly Property Item(ByVal name As String) As

1	OperationFault
2	[JScript] returnValue = OperationFaultCollectionObject.Item(name);
3	
4	Description
5	Gets an System. Web. Services. Description. Operation Fault by its
6	System.Web.Services.Description.OperationFault.Name property. The string
7	value representing the name of the
8	System.Web.Services.Description.OperationFault returned.
9	List
10	Table
11	Add
12	
13	[C#] public int Add(OperationFault operationFaultMessage);
14	[C++] public: int Add(OperationFault* operationFaultMessage);
15	[VB] Public Function Add(ByVal operationFaultMessage As OperationFault) As
16	Integer
17	[JScript] public function Add(operationFaultMessage : OperationFault) : int;
18	
19	Description
20	Adds the specified System. Web. Services. Description. Operation Fault to
21	the end of the OperationFaultCollection.
22	Return Value: Returns the zero-based index where operationFaultMessage has
23	been added. The System.Web.Services.Description.OperationFault to be added
24	to the OperationFaultCollection.
25	Contains

1	
2	[C#] public bool Contains(OperationFault operationFaultMessage);
3	[C++] public: bool Contains(OperationFault* operationFaultMessage);
4	[VB] Public Function Contains(ByVal operationFaultMessage As OperationFault)
5	As Boolean
6	[JScript] public function Contains(operationFaultMessage : OperationFault) :
7	Boolean;
8	
9	Description
10	Gets a value indicating whether the specified
11	System.Web.Services.Description.OperationFault is a member of the
12	OperationFaultCollection .
13	Return Value: true if operationFaultMessage is a member of the
14	OperationFaultCollection; otherwise, false. An
15	System.Web.Services.Description.OperationFault object.
16	СоруТо
17	
18	[C#] public void CopyTo(OperationFault[] array, int index);
19	[C++] public: void CopyTo(OperationFault* array[], int index);
20	[VB] Public Sub CopyTo(ByVal array() As OperationFault, ByVal index As
21	Integer)
22	[JScript] public function CopyTo(array: OperationFault[], index: int);
23	
24	Description
25	

9

10

11

1

12 13

17

18

15

16

19

21

20

22

2425

Copies the entire

System.Web.Services.Description.OperationFaultCollection to a compatible one-dimensional array, starting at the specified zero-based index of the target array. An array of type System.Web.Services.Description.OperationFault serving as the destination of the copy action. The zero-based index at which to start placing the copied collection.

GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String [JScript] protected override function GetKey(value : Object) : String;

Description

Returns the name of the

**System.Web.Services.Description.OperationFault** associated with the value passed by reference. An object for which to return the name.

IndexOf

[C#] public int IndexOf(OperationFault operationFaultMessage);

[C++] public: int IndexOf(OperationFault\* operationFaultMessage);

[VB] Public Function IndexOf(ByVal operationFaultMessage As OperationFault)

As Integer

[JScript] public function IndexOf(operationFaultMessage : OperationFault) : int;

557

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Searches for the specified

**System.Web.Services.Description.OperationFault** and returns the zero-based index of the first occurrence within the collection.

Return Value: Returns a 32-bit signed integer. An

System. Web. Services. Description. Operation Fault object.

Insert

[C#] public void Insert(int index, OperationFault operationFaultMessage);

[C++] public: void Insert(int index, OperationFault\* operationFaultMessage);

[VB] Public Sub Insert(ByVal index As Integer, ByVal operationFaultMessage As

OperationFault)

[JScript] public function Insert(index: int, operationFaultMessage:

OperationFault);

Description

Adds the specified **System.Web.Services.Description.OperationFault** to the **OperationFaultCollection** at the specified zero-based index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert operationFaultMessage. The System.Web.Services.Description.OperationFault to be added to the Collection.

Remove

1	
2	[C#] public void Remove(OperationFault operationFaultMessage);
3	[C++] public: void Remove(OperationFault* operationFaultMessage);
4	[VB] Public Sub Remove(ByVal operationFaultMessage As OperationFault)
5	[JScript] public function Remove(operationFaultMessage : OperationFault);
6	
7	Description
8	Removes the first occurrence of the specified
9	System.Web.Services.Description.OperationFault from the
10	OperationFaultCollection .
11	This method performs a linear search; therefore, the average execution time
12	is proportional to
13	System. We b. Services. Description. Operation Fault Collection. Count. The
14	System.Web.Services.Description.OperationFault object to be removed from
15	the collection.
16	SetParent
17	
18	[C#] protected override void SetParent(object value, object parent);
19	[C++] protected: void SetParent(Object* value, Object* parent);
20	[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
21	Object)
22	[JScript] protected override function SetParent(value : Object, parent : Object);
23	
24	Description
25	

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Sets the parent **System.Web.Services.Description.Operation** of a member of the **OperationFaultCollection**. An object, of class **System.Web.Services.Description.OperationFault**, within the collection. The object, of class **System.Web.Services.Description.Operation**, to be set as the parent.

OperationFlow enumeration (System.Web.Services.Description)
ToString

# Description

Specifies the type of transmission an endpoint of the Web Service can support.

Although Request-Response or Solicit-Response are logically correlated in the WSDL document, the concrete correlation information will be specified by a binding. For example, the request and response messages may be exchanged as part of one or two actual HTTP transmissions.

**ToString** 

[C#] public const OperationFlow None;

[C++] public: const OperationFlow None;

[VB] Public Const None As OperationFlow

[JScript] public var None : OperationFlow;

# Description

Indicates that the endpoint of the Web Service receives no transmissions.

	1	ToString
	2	
	3	[C#] public const OperationFlow Notification;
	4	[C++] public: const OperationFlow Notification;
	5	[VB] Public Const Notification As OperationFlow
	6	[JScript] public var Notification : OperationFlow;
	7	
	8	Description
	9	Indicates that an endpoint of the Web Service sends a message.
nail.	10	ToString
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	11	
, mm,	12	[C#] public const OperationFlow OneWay;
	13	[C++] public: const OperationFlow OneWay;
1	14	[VB] Public Const OneWay As OperationFlow
	15	[JScript] public var OneWay : OperationFlow;
	16	
•	17	Description
	18	Indicates that the endpoint of the Web Service receives a message
	19	ToString
	20	
	21	[C#] public const OperationFlow RequestResponse;
	22	[C++] public: const OperationFlow RequestResponse;
	23	[VB] Public Const RequestResponse As OperationFlow
	24	[JScript] public var RequestResponse : OperationFlow;
	25	

### Description

1

3

5

6

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Indicates that an endpoint of the Web Service receives a message, then sends a correlated message.

**ToString** 

[C#] public const OperationFlow SolicitResponse;

[C++] public: const OperationFlow SolicitResponse;

[VB] Public Const SolicitResponse As OperationFlow

[JScript] public var SolicitResponse : OperationFlow;

### Description

Indicates that an endpoint of the Web Service sends a message, then receives a correlated message.

OperationInput class (System.Web.Services.Description)

**ToString** 

## Description

Defines the specifications for input messages received by the Web Service.

This class cannot be inherited.

Exactly one instance of this class will be a member of the

System.Web.Services.Description.Operation.Messages property of the parent System.Web.Services.Description.Operation.

stem. Web. Set vices. Description. Oper

OperationInput

1	Example Syntax:
2	ToString
3	
4	[C#] public OperationInput();
5	[C++] public: OperationInput();
6	[VB] Public Sub New()
7	[JScript] public function OperationInput();
8	Documentation
9	Message
10	Name
11	Operation
12	OperationMessage class (System.Web.Services.Description)
13	ToString
14	
15	
16	Description
17	Serves as the base class for the following classes:
18	System.Web.Services.Description.OperationFaultSystem.Web.Services.Descri
19	ption.OperationInputSystem.Web.Services.Description.OperationOutput
20	In its default implementation, an <b>OperationMessage</b> instance refers to any
21	message type passed by the action of a service.
22	OperationMessage
23	Example Syntax:
24	ToString
25	

```
1
   [C#] protected OperationMessage();
   [C++] protected: OperationMessage();
   [VB] Protected Sub New()
   [JScript] protected function OperationMessage();
5
           Documentation
6
           Message
           ToString
8
9
10
    Description
11
           Gets or sets an abstract, typed definition of the data being communicated.
12
           Name
13
           ToString
14
15
    [C#] public string Name {get; set;}
16
    [C++] public: __property String* get_Name();public: __property void
17
    set Name(String*);
18
    [VB] Public Property Name As String
19
    [JScript] public function get Name(): String; public function set Name(String);
20
21
    Description
22
            The name of the System. Web. Services. Description. Operation Message
23
     instance.
24
            Operation
25
```

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

	_ ~		
11	oS	***1	110
	(),		112
_	$\sim$		

[C#] public Operation Operation {get;}

[C++] public: \_\_property Operation\* get\_Operation();

[VB] Public ReadOnly Property Operation As Operation

[JScript] public function get Operation(): Operation;

# Description

Gets the System.Web.Services.Description.Operation of which the System.Web.Services.Description.OperationMessage instance is a member.

OperationMessageCollection class (System.Web.Services.Description)
ToString

## Description

Represents a collection of

System. Web. Services. Description. Operation Input and

System.Web.Services.Description.OperationOutput messages related to a Web Service. This class cannot be inherited.

An instance of this class will be returned by the

System.Web.Services.Description.Operation.Messages property of the parent System.Web.Services.Description.Operation . As such, it can have exactly two members, one an System.Web.Services.Description.OperationInput and the other an System.Web.Services.Description.OperationOutput .

Count

Flow **ToString** 2 3 Description 5 Gets the type of transmission supported by the 6  $Operation Message Collection \ .$ 7 InnerList 8 Input 9 **ToString** 10 11 12 Description 13 Gets the first occurrence of an 14 System. Web. Services. Description. Operation Input instance within the 15 collection. 16 Th 17 Item 18 **ToString** 19 20 [C#] public OperationMessage this[int index] {get; set;} 21 [C++] public: \_\_property OperationMessage\* get\_Item(int index);public: 22 \_property void set\_Item(int index, OperationMessage\*); 23 [VB] Public Default Property Item(ByVal index As Integer) As OperationMessage 24 [JScript] returnValue =

OperationMessageCollectionObject.Item(index);OperationMessageCollectionObje ct.Item(index) = returnValue; 3 Description Gets or sets the value of an System.Web.Services.Description.OperationMessage at the specified zerobased index. The zero-based index of the System.Web.Services.Description.OperationMessage whose value is modified or returned. List 10 Output 11 **ToString** 12 13 14 Description 15 Gets the first occurrence of an 16 System. Web. Services. Description. Operation Output within the collection. 17 The search is performed in index order; thus the property returns only the 18 System.Web.Services.Description.OperationOutput with the lowest-numbered 19 index. 20 Table 21 Add 22 23 [C#] public int Add(OperationMessage operationMessage); 24 [C++] public: int Add(OperationMessage\* operationMessage);

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[VB] Public Function Add(ByVal operationMessage As OperationMessage) As Integer [JScript] public function Add(operationMessage : OperationMessage) : int; Description Adds the specified System. Web. Services. Description. Operation Message to the end of the  ${\bf Operation Message Collection}$  . Return Value: Returns the zero-based index where operationMessage has been added. The System.Web.Services.Description.OperationMessage to be added to the collection. Contains [C#] public bool Contains(OperationMessage operationMessage); [C++] public: bool Contains(OperationMessage\* operationMessage); [VB] Public Function Contains(ByVal operationMessage As OperationMessage) As Boolean [JScript] public function Contains(operationMessage : OperationMessage) : Boolean; Description Gets a value indicating whether the specified System.Web.Services.Description.OperationMessage instance is a member of  $the \ Operation Message Collection \ .$ Return Value: true if operationMessage is a member of the

OperationMessageCollection; otherwise, false. An 1 System. Web. Services. Description. Operation Message object. 2 CopyTo 3 4 [C#] public void CopyTo(OperationMessage[] array, int index); 5 [C++] public: void CopyTo(OperationMessage\* array[], int index); [VB] Public Sub CopyTo(ByVal array() As OperationMessage, ByVal index As 7 Integer) 8 [JScript] public function CopyTo(array: OperationMessage[], index: int); 9 10 Description 11 Copies the entire OperationMessageCollection to a compatible one-12 dimensional array of type System. Web. Services. Description. Operation Message 13 , starting at the specified zero-based index of the target array. An array of type 14 System. Web. Services. Description. Operation Message serving as the destination 15 of the copy action. The zero-based index at which to start placing the copied 16 collection. 17 IndexOf 18 19 [C#] public int IndexOf(OperationMessage operationMessage); 20 [C++] public: int IndexOf(OperationMessage\* operationMessage); 21 [VB] Public Function IndexOf(ByVal operationMessage As OperationMessage) 22 As Integer 23 [JScript] public function IndexOf(operationMessage : OperationMessage) : int; 24 25

-		, •
Des	orin	TINK
Desi	$\sim \iota \nu$	uvi

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Searches for the specified OperationMessage instance and returns the zero-based index of the first occurrence within the **OperationMessageCollection** .

Return Value: Returns a 32-bit signed integer. An

System. Web. Services. Description. Operation Message object.

Insert

[C#] public void Insert(int index, OperationMessage operationMessage);

[C++] public: void Insert(int index, OperationMessage\* operationMessage);

[VB] Public Sub Insert(ByVal index As Integer, ByVal operationMessage As

OperationMessage)

[JScript] public function Insert(index: int, operationMessage:

OperationMessage);

### Description

Adds the specified **System.Web.Services.Description.OperationMessage** to the **OperationMessageCollection** at the specified zero-based index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert operationMessage. The System.Web.Services.Description.OperationMessage to be added to the OperationMessageCollection.

OnInsert

2	[C#] protected override void OnInsert(int index, object value);
3	[C++] protected: void OnInsert(int index, Object* value);
4	[VB] Overrides Protected Sub OnInsert(ByVal index As Integer, ByVal value As
5	Object)
6	[JScript] protected override function OnInsert(index : int, value : Object);
7	
8	Description
9	Performs validation upon insertion of an
10	System.Web.Services.Description.OperationMessage at the specified specified.
11	This method overrides the base method in order to limit the number of
12	collection members to two (an
13	System.Web.Services.Description.OperationInput and an
14	System.Web.Services.Description.OperationOutput). The zero-based index at
15	which to insert the specified
16	System.Web.Services.Description.OperationMessage. The
17	System.Web.Services.Description.OperationMessage to be added to the
18	OperationMessageCollection.
19	OnSet
20	
21	[C#] protected override void OnSet(int index, object oldValue, object newValue);
22	[C++] protected: void OnSet(int index, Object* oldValue, Object* newValue);
23	[VB] Overrides Protected Sub OnSet(ByVal index As Integer, ByVal oldValue As
24	Object, ByVal newValue As Object)

[JScript] protected override function OnSet(index : int, oldValue : Object,

newValue: Object);

2

Description

Op

7

8

9

10

12

14

16 17

18

19 20

21

23

24

25

Description

Performs validation upon replacement of the specified member of the OperationMessageCollection with a new OperationMessage at the specified zerobased index.

This method overrides the base method in order to assure that the value replacing the existing value is of the same class as that existing value. Thus an **System.Web.Services.Description.OperationInput** must be replaced by an **OperationInput**, and an **System.Web.Services.Description.OperationOutput** must be replaced by an **OperationOutput**. The zero-based index at which to change *oldValue* to *newValue*. The object value to be replaced by *newValue*. The object value with which to replace *oldValue*.

OnValidate

[C#] protected override void OnValidate(object value);

[C++] protected: void OnValidate(Object\* value);

[VB] Overrides Protected Sub OnValidate(ByVal value As Object)

[JScript] protected override function OnValidate(value : Object);

Description

Performs a check on the type of the specified object when validating it.

This method overrides the base method in order to assure that the instance represented by the *value* parameter is either of type

System.Web.Services.Description.OperationInput or

1	System.Web.Services.Description.OperationOutput. The object to be
2	validated.
3	Remove
4	
5	[C#] public void Remove(OperationMessage operationMessage);
6	[C++] public: void Remove(OperationMessage* operationMessage);
7	[VB] Public Sub Remove(ByVal operationMessage As OperationMessage)
8	[JScript] public function Remove(operationMessage : OperationMessage);
9	
10	Description
11	Removes the first occurrence of the specified
12	System.Web.Services.Description.OperationMessage from the
13	OperationMessageCollection .
14	This method performs a linear search; therefore, the average execution time
15	is proportional to Count . An
16	System.Web.Services.Description.OperationMessage object.
17	SetParent
18	
19	[C#] protected override void SetParent(object value, object parent);
20	[C++] protected: void SetParent(Object* value, Object* parent);
21	[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
22	Object)
23	[JScript] protected override function SetParent(value : Object, parent : Object);
24	
25	Description

1	Sets the parent System. Web. Services. Description. Operation of the
2	OperationMessageCollection . An object, of type
3	System.Web.Services.Description.OperationMessage or a derived type, within
4	the collection. The object, of type System.Web.Services.Description.Operation
5	to be set as the parent.
6	OperationOutput class (System.Web.Services.Description)
7	ToString
8	
9	
10	Description
11	Defines the specifications for output messages returned by the Web
12	Service. This class cannot be inherited.
13	Exactly one instance of this class will be a member of the
14	System.Web.Services.Description.Operation.Messages property of the parent
15	System.Web.Services.Description.Operation .
16	OperationOutput
17	Example Syntax:
18	ToString
19	
20	[C#] public OperationOutput();
21	[C++] public: OperationOutput();
22	[VB] Public Sub New()
23	[JScript] public function OperationOutput();
24	Documentation
25	Message

1	Name
2	Operation
3	OutputBinding class (System.Web.Services.Description)
4	ToString
5	
6	
7	Description
8	Provides a set of specifications for data formats and protocols used by the
9	Web Service for output messages. This class cannot be inherited.
10	OutputBinding
11	Example Syntax:
12	ToString
13	
14	[C#] public OutputBinding();
15	[C++] public: OutputBinding();
16	[VB] Public Sub New()
17	[JScript] public function OutputBinding();
18	Documentation
19	Extensions
20	ToString
21	
22	
23	Description
24	
25	

	7		ts	- 4 '	١.	
ı		Ο.	rc.	Т	n	6

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect ion associated with the OutputBinding instance.

Name

**OperationBinding** 

Port class (System.Web.Services.Description)

**ToString** 

## Description

Defines an individual endpoint contained in the Web Service. This class cannot be inherited.

This class specifies a single address (URI) for a binding, which defines message format and protocol details for operations and messages for a given System.Web.Services.Description.PortType. The

**System.Web.Services.Description.Port.Binding** property gets or sets those values for the **Port** instance.

Port

Example Syntax:

**ToString** 

[C#] public Port();

[C++] public: Port();

[VB] Public Sub New()

[JScript] public function Port();

1	Binding
2	ToString
3	
4	[C#] public XmlQualifiedName Binding {get; set;}
5	[C++] public:property XmlQualifiedName* get_Binding();public:property
6	<pre>void set_Binding(XmlQualifiedName*);</pre>
7	[VB] Public Property Binding As XmlQualifiedName
8	[JScript] public function get Binding(): XmlQualifiedName; public function set
9	Binding(XmlQualifiedName);
10	
11	Description
12	Gets or sets the value of the XML binding attribute of the Port.
13	A binding defines message format and protocol details for operations and
14	messages for a given System.Web.Services.Description.PortType . This
15	property gets or sets those values for a specific Port.
16	Documentation
17	Extensions
18	ToString
19	
20	
21	Description
22	Gets the collection of extensibility elements contained in the Port.
23	Name
24	ToString
25	

1	
2	[C#] public string Name {get; set;}
3	[C++] public:property String* get_Name();public:property void
4	set_Name(String*);
5	[VB] Public Property Name As String
6	[JScript] public function get Name(): String; public function set Name(String);
7	
8	Description
9	Gets or sets the Name of the Port.
10	Service
11	ToString
12	
13	[C#] public Service Service {get;}
14	[C++] public:property Service* get_Service();
15	[VB] Public ReadOnly Property Service As Service
16	[JScript] public function get Service() : Service;
17	
18	Description
19	Gets the System.Web.Services.Description.Service of which the Port
20	instance is a member.
21	PortCollection class (System.Web.Services.Description)
22	ToString
23	
24	
ا ء د	Description

Represents the collection of all System. Web. Services. Description. Port 1 objects contained within a System. Web. Services. Description. Service. This class 2 cannot be inherited. 3 Count 4 InnerList 5 Item 6 **ToString** 7 System.Web.Services.Description.Port 8 9 Description 10 Gets or sets the value of a System. Web. Services. Description. Port at the 11 specified zero-based index. The zero-based index of the 12 **System.Web.Services.Description.Port** whose value is modified or returned. 13 Item 14 **ToString** 15 16 [C#] public Port this[string name] {get;} 17 [C++] public: property Port\* get Item(String\* name); 18 [VB] Public Default ReadOnly Property Item(ByVal name As String) As Port 19 [JScript] returnValue = PortCollectionObject.Item(name); 20 21 Description 22 Gets the specified System. Web. Services. Description. Port by its 23 System.Web.Services.Description.Port.Name property. A string value 24 representing the name of the System. Web. Services. Description. Port returned.

1	List
2	Table
3	Add
4	
5	[C#] public int Add(Port port);
6	[C++] public: int Add(Port* port);
7	[VB] Public Function Add(ByVal port As Port) As Integer
8	[JScript] public function Add(port : Port) : int;
9	
10	Description
11	Adds the specified System. Web. Services. Description. Port instance to the
12	end of the PortCollection.
13	Return Value: Returns the index where the specified
14	System.Web.Services.Description.Port instance has been added. The
15	System.Web.Services.Description.Port to be added to the PortCollection.
16	Contains
17	
18	[C#] public bool Contains(Port port);
19	[C++] public: bool Contains(Port* port);
20	[VB] Public Function Contains(ByVal port As Port) As Boolean
21	[JScript] public function Contains(port : Port) : Boolean;
22	
23	Description
24	Gets a value indicating whether the specified
25	System.Web.Services.Description.Port instance is a member of the

1	PortCollection .
2	Return Value: true if the specified System.Web.Services.Description.Port is a
3	member of the PortCollection; otherwise, false. A
4	System.Web.Services.Description.Port object.
5	СоруТо
6	
7	[C#] public void CopyTo(Port[] array, int index);
8	[C++] public: void CopyTo(Port* array[], int index);
9	[VB] Public Sub CopyTo(ByVal array() As Port, ByVal index As Integer)
10	[JScript] public function CopyTo(array: Port[], index: int);
11	
12	Description
13	Copies the entire System.Web.Services.Description.PortCollection to a
14	one-dimensional array of type System. Web. Services. Description. Port, starting
15	at the specified zero-based index of the target array. An array of type
16	System.Web.Services.Description.Port serving as the destination of the copy
17	action. The zero-based index at which to start placing the copied PortCollection
18	GetKey
19	
20	[C#] protected override string GetKey(object value);
21	[C++] protected: String* GetKey(Object* value);
22	[VB] Overrides Protected Function GetKey(ByVal value As Object) As String
23	[JScript] protected override function GetKey(value : Object) : String;
24	
25	Description

1	Returns the name of the System. Web. Services. Description. Port instance
2	associated with the value passed by reference. A
3	System.Web.Services.Description.Port instance for which to return the name.
4	IndexOf
5	
6	[C#] public int IndexOf(Port port);
7	[C++] public: int IndexOf(Port* port);
8	[VB] Public Function IndexOf(ByVal port As Port) As Integer
9	[JScript] public function IndexOf(port : Port) : int;
10	
11	Description
12	Searches for the specified System.Web.Services.Description.Port
13	instance and returns the zero-based index of the first occurrence within the
14	PortCollection .
15	Return Value: Returns a 32-bit signed integer. A
16	System.Web.Services.Description.Port object.
17	Insert
18	
19	[C#] public void Insert(int index, Port port);
20	[C++] public: void Insert(int index, Port* port);
21	[VB] Public Sub Insert(ByVal index As Integer, ByVal port As Port)
22	[JScript] public function Insert(index: int, port: Port);
23	
24	Description
25	

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Adds the specified **System.Web.Services.Description.Port** instance to the **PortCollection** at the specified index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert *port*. The **System.Web.Services.Description.Port** to be added to the collection.

Remove

[C#] public void Remove(Port port);

[C++] public: void Remove(Port\* port);

[VB] Public Sub Remove(ByVal port As Port)

[JScript] public function Remove(port : Port);

Description

Removes the first occurrence of the specified

System. Web. Services. Description. Port from the PortCollection.

This method performs a linear search; therefore, the average execution time is proportional to **Count**. A **System.Web.Services.Description.Port** object.

SetParent

[C#] protected override void SetParent(object value, object parent);

[C++] protected: void SetParent(Object\* value, Object\* parent);

[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As

Object)

[JScript] protected override function SetParent(value : Object, parent : Object);

_		_	_	
71	escr		tin:	•
,,,	$\nu \times c r$	777	,,,,,	r

3

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Sets the parent System.Web.Services.Description.Service of a member of the PortCollection . An object, of type System.Web.Services.Description.Port, within the collection. The object, of type

**System.Web.Services.Description.Service**, to set as the parent.

PortType class (System.Web.Services.Description)

**ToString** 

### Description

Represents a named set of abstract operations and the corresponding abstract messages. This class cannot be inherited.

PortType

Example Syntax:

**ToString** 

[C#] public PortType();

[C++] public: PortType();

[VB] Public Sub New()

[JScript] public function PortType();

Documentation

Name

**ToString** 

1	
2	
3	Description
4	Gets or sets the name of the System.Web.Services.Description.PortType
5	instance.
6	Returns an empty string ("") if the property value has not been assigned.
7	Operations
8	ToString
9	
10	[C#] public OperationCollection Operations {get;}
11	[C++] public:property OperationCollection* get_Operations();
12	[VB] Public ReadOnly Property Operations As OperationCollection
13	[JScript] public function get Operations(): OperationCollection;
14	
15	Description
16	Gets the collection of System.Web.Services.Description.Operation
17	instances defined by the System. Web. Services. Description. Port Type instance.
18	ServiceDescription
19	ToString
20	
21	[C#] public ServiceDescription ServiceDescription {get;}
22	[C++] public:property ServiceDescription* get_ServiceDescription();
23	[VB] Public ReadOnly Property ServiceDescription As ServiceDescription
24	[JScript] public function get ServiceDescription(): ServiceDescription;
25	

40
ıI
1 100
ï
i i
H
, <sup>, , ,</sup> ,
ļ. Ja
lı all

T .		. •
1100	crin	tint
1765		LLUIT

Gets the **System.Web.Services.Description.ServiceDescription** of which the **System.Web.Services.Description.PortType** instance is a member.

PortTypeCollection class (System.Web.Services.Description)

**ToString** 

# Description

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Represents a collection of **System.Web.Services.Description.PortType** objects, that is, a collection of sets of operations supported by the Web Service. This class cannot be inherited.

Count

InnerList

Item

**ToString** 

System.Web.Services.Description.PortType

# Description

Gets or sets the value of a **System.Web.Services.Description.PortType** at the specified zero-based index. The zero-based index of the **System.Web.Services.Description.PortType** whose value is modified or returned.

Item

11011

**ToString** 

```
1
    [C#] public PortType this[string name] {get;}
    [C++] public: property PortType* get Item(String* name);
    [VB] Public Default ReadOnly Property Item(ByVal name As String) As
    PortType
    [JScript] returnValue = PortTypeCollectionObject.Item(name);
 7
    Description
 8
           Gets the specified System. Web. Services. Description. Port Type by its
 9
    System.Web.Services.Description.PortType.Name property.
10
           This method performs a linear search; therefore, the average execution time
11
    is proportional to System. Web. Services. Description. Port Type Collection. Count
12
    . The string value representing the name of the
13
    System.Web.Services.Description.PortType returned.
14
           List
15
           Table
16
           Add
17
18
    [C#] public int Add(PortType portType);
19
    [C++] public: int Add(PortType* portType);
20
    [VB] Public Function Add(ByVal portType As PortType) As Integer
21
    [JScript] public function Add(portType : PortType) : int;
22
23
    Description
24
25
```

1	Adds the specified System.Web.Services.Description.PortType to the end
2	of the PortTypeCollection.
3	Return Value: Returns the index where the
4	System.Web.Services.Description.PortType instance specified by the portType
5	parameter has been added. The System.Web.Services.Description.PortType to
6	be added to the PortTypeCollection.
7	Contains
8	
9	[C#] public bool Contains(PortType portType);
10	[C++] public: bool Contains(PortType* portType);
11	[VB] Public Function Contains(ByVal portType As PortType) As Boolean
12	[JScript] public function Contains(portType : PortType) : Boolean;
13	
14	Description
15	Gets a value indicating whether the specified
16	System.Web.Services.Description.PortType instance is a member of the
17	PortTypeCollection .
18	Return Value: true if the specified System.Web.Services.Description.PortType
19	instance is a member of the PortTypeCollection; otherwise, false. A
20	System.Web.Services.Description.PortType object.
21	СоруТо
22	
23	[C#] public void CopyTo(PortType[] array, int index);
24	[C++] public: void CopyTo(PortType* array[], int index);
25	[VB] Public Sub CopyTo(ByVal array() As PortType, ByVal index As Integer)

21

22

23

24

[JScript] public function CopyTo(array : PortType[], index : int);

Description

Copies the entire **System.Web.Services.Description.PortTypeCollection** to a one-dimensional array of type PortType, starting at the specified zero-based index of the target array. An array of type

**System.Web.Services.Description.PortType** serving as the destination of the copy action. The zero-based index at which to start placing the copied **PortTypeCollection**.

GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String [JScript] protected override function GetKey(value : Object) : String;

Description

Returns the name of the **System.Web.Services.Description.PortType** associated with the value passed by reference. An object for which to return the name.

IndexOf

[C#] public int IndexOf(PortType portType);

[C++] public: int IndexOf(PortType\* portType);

[VB] Public Function IndexOf(ByVal portType As PortType) As Integer

24

25

[JScript] public function IndexOf(portType : PortType) : int;

Description

1

Searches for the specified **System.Web.Services.Description.PortType** instance and returns the zero-based index of the first occurrence within the **PortTypeCollection**.

Return Value: Returns a 32-bit signed integer. A

System.Web.Services.Description.PortType object.

Insert

[C#] public void Insert(int index, PortType portType);

[C++] public: void Insert(int index, PortType\* portType);

[VB] Public Sub Insert(ByVal index As Integer, ByVal portType As PortType)

[JScript] public function Insert(index : int, portType : PortType);

Description

Adds the specified **System.Web.Services.Description.PortType** instance to the **PortTypeCollection** at the specified zero-based index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert the specified System.Web.Services.Description.PortType instance. The System.Web.Services.Description.PortType to be added to the PortTypeCollection.

Remove

1	
2	[C#] public void Remove(PortType portType);
3	[C++] public: void Remove(PortType* portType);
4	[VB] Public Sub Remove(ByVal portType As PortType)
5	[JScript] public function Remove(portType : PortType);
6	
7	Description
8	Removes the first occurrence of the specified
9	System.Web.Services.Description.PortType from the PortTypeCollection .
10	This method performs a linear search; therefore, the average execution time
11	is proportional to Count . A System. Web. Services. Description. PortType object.
12	SetParent
13	
14	[C#] protected override void SetParent(object value, object parent);
15	[C++] protected: void SetParent(Object* value, Object* parent);
16	[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
17	Object)
18	[JScript] protected override function SetParent(value : Object, parent : Object);
19	
20	Description
21	Sets the parent System. Web. Services. Description. Service Description of
22	a member of System. Web. Services. Description. Port Type Collection . An object
23	of type System.Web.Services.Description.PortType, within the
24	PortTypeCollection . The object, of type
25	System.Web.Services.Description.ServiceDescription, to set as the parent.

1	ProtocolImporter class (System.Web.Services.Description)
2	ToString
3	
4	
5	Description
6	
7	ProtocolImporter
8	Example Syntax:
9	ToString
10	
11	[C#] protected ProtocolImporter();
12	[C++] protected: ProtocolImporter();
13	[VB] Protected Sub New()
14	[JScript] protected function ProtocolImporter();
15	AbstractSchemas
16	ToString
17	
18	[C#] public XmlSchemas AbstractSchemas {get;}
19	[C++] public:property XmlSchemas* get_AbstractSchemas();
20	[VB] Public ReadOnly Property AbstractSchemas As XmlSchemas
21	[JScript] public function get AbstractSchemas(): XmlSchemas;
22	
23	Description
24	
25	Binding

1	ToString
2	
3	[C#] public Binding Binding {get;}
4	[C++] public:property Binding* get_Binding();
5	[VB] Public ReadOnly Property Binding As Binding
6	[JScript] public function get Binding() : Binding;
7	
8	Description
9	
10	ClassName
11	ToString
12	
13	[C#] public string ClassName {get;}
14	[C++] public:property String* get_ClassName();
15	[VB] Public ReadOnly Property ClassName As String
16	[JScript] public function get ClassName() : String;
17	
18	Description
19	
20	ClassNames
21	ToString
22	
23	[C#] public CodeIdentifiers ClassNames {get;}
24	[C++] public:property CodeIdentifiers* get_ClassNames();
25	[VB] Public ReadOnly Property ClassNames As CodeIdentifiers

1	[JScript] public function get ClassNames(): CodeIdentifiers;
2	
3	Description
4	
5	CodeNamespace
6	ToString
7	
8	[C#] public CodeNamespace CodeNamespace {get;}
9	[C++] public:property CodeNamespace* get_CodeNamespace();
10	[VB] Public ReadOnly Property CodeNamespace As CodeNamespace
11	[JScript] public function get CodeNamespace() : CodeNamespace;
12	
13	Description
14	
15	CodeTypeDeclaration
16	ToString
17	
18	[C#] public CodeTypeDeclaration CodeTypeDeclaration {get;}
19	[C++] public:property CodeTypeDeclaration* get_CodeTypeDeclaration();
20	[VB] Public ReadOnly Property CodeTypeDeclaration As CodeTypeDeclaration
21	[JScript] public function get CodeTypeDeclaration(): CodeTypeDeclaration;
22	
23	Description
24	
25	ConcreteSchemas

1	ToString
2	
3	[C#] public XmlSchemas ConcreteSchemas {get;}
4	[C++] public:property XmlSchemas* get_ConcreteSchemas();
5	[VB] Public ReadOnly Property ConcreteSchemas As XmlSchemas
6	[JScript] public function get ConcreteSchemas(): XmlSchemas;
7	
8	Description
9	
10	InputMessage
11	ToString
12	
13	[C#] public Message InputMessage {get;}
14	[C++] public:property Message* get_InputMessage();
15	[VB] Public ReadOnly Property InputMessage As Message
16	[JScript] public function get InputMessage() : Message;
17	
18	Description
19	
20	MethodName
21	ToString
22	
23	[C#] public string MethodName {get;}
24	[C++] public:property String* get_MethodName();
25	[VB] Public ReadOnly Property MethodName As String

1	[JScript] public function get MethodName(): String;
2	
3	Description
4	
5	Operation
6	ToString
7	
8	[C#] public Operation (get;)
9	[C++] public:property Operation* get_Operation();
10	[VB] Public ReadOnly Property Operation As Operation
11	[JScript] public function get Operation(): Operation;
12	
13	Description
14	
15	OperationBinding
16	ToString
17	
18	[C#] public OperationBinding OperationBinding {get;}
19	[C++] public:property OperationBinding* get_OperationBinding();
20	[VB] Public ReadOnly Property OperationBinding As OperationBinding
21	[JScript] public function get OperationBinding(): OperationBinding;
22	
23	Description
24	
25	OutputMessage

```
ToString
2
    [C#] public Message OutputMessage {get;}
3
    [C++] public: __property Message* get_OutputMessage();
    [VB] Public ReadOnly Property OutputMessage As Message
5
    [JScript] public function get OutputMessage(): Message;
7
    Description
8
9
           Port
10
           ToString
11
12
    [C#] public Port Port {get;}
13
    [C++] public: property Port* get Port();
14
    [VB] Public ReadOnly Property Port As Port
15
    [JScript] public function get Port(): Port;
16
17
    Description
18
19
           PortType
20
           ToString
21
22
    [C#] public PortType PortType {get;}
23
    [C++] public: __property PortType* get_PortType();
24
    [VB] Public ReadOnly Property PortType As PortType
25
```

```
[JScript] public function get PortType(): PortType;
 2
    Description
 3
           ProtocolName
           ToString
 7
    [C#] public abstract string ProtocolName {get;}
 8
    [C++] public: __property virtual String* get ProtocolName() = 0;
 9
    [VB] MustOverride Public ReadOnly Property ProtocolName As String
10
    [JScript] public abstract function get ProtocolName(): String;
11
12
    Description
13
14
           Schemas
15
           ToString
16
17
    [C#] public XmlSchemas Schemas {get;}
18
    [C++] public: property XmlSchemas* get Schemas();
19
    [VB] Public ReadOnly Property Schemas As XmlSchemas
20
    [JScript] public function get Schemas(): XmlSchemas;
21
22
    Description
23
24
           Service
25
```

1	ToString
2	
3	[C#] public Service Service {get;}
4	[C++] public:property Service* get_Service();
5	[VB] Public ReadOnly Property Service As Service
6	[JScript] public function get Service() : Service;
7	
8	Description
9	
10	ServiceDescriptions
11	ToString
12	
13	[C#] public ServiceDescriptionCollection ServiceDescriptions {get;}
14	[C++] public:property ServiceDescriptionCollection*
15	get_ServiceDescriptions();
16	[VB] Public ReadOnly Property ServiceDescriptions As
17	ServiceDescriptionCollection
18	[JScript] public function get ServiceDescriptions(): ServiceDescriptionCollection;
19	
20	Description
21	
22	Style
23	ToString
24	
25	[C#] public ServiceDescriptionImportStyle Style {get;}

1	[C++] public:property ServiceDescriptionImportStyle get_Style();
2	[VB] Public ReadOnly Property Style As ServiceDescriptionImportStyle
3	[JScript] public function get Style() : ServiceDescriptionImportStyle;
4	
5	Description
6	
7	Warnings
8	ToString
9	
10	[C#] public ServiceDescriptionImportWarnings Warnings {get; set;}
11	[C++] public:property ServiceDescriptionImportWarnings
12	get_Warnings();public:property void
13	set_Warnings(ServiceDescriptionImportWarnings);
14	[VB] Public Property Warnings As ServiceDescriptionImportWarnings
15	[JScript] public function get Warnings():
16	ServiceDescriptionImportWarnings;public function set
17	Warnings(ServiceDescriptionImportWarnings);
18	
19	Description
20	
21	AddExtensionWarningComments
22	
23	[C#] public void
24	AddExtensionWarningComments(CodeCommentStatementCollection comments
25	ServiceDescriptionFormatExtensionCollection extensions);

1	[C++] public: void
2	AddExtensionWarningComments(CodeCommentStatementCollection* comments
3	ServiceDescriptionFormatExtensionCollection* extensions);
4	[VB] Public Sub AddExtensionWarningComments(ByVal comments As
5	CodeCommentStatementCollection, ByVal extensions As
6	ServiceDescriptionFormatExtensionCollection)
7	[JScript] public function AddExtensionWarningComments(comments:
8	CodeCommentStatementCollection, extensions:
9	ServiceDescriptionFormatExtensionCollection);
10	
11	Description
12	
13	BeginClass
14	
15	[C#] protected abstract CodeTypeDeclaration BeginClass();
16	[C++] protected: virtual CodeTypeDeclaration* BeginClass() = 0;
17	[VB] MustOverride Protected Function BeginClass() As CodeTypeDeclaration
18	[JScript] protected abstract function BeginClass(): CodeTypeDeclaration;
19	
20	Description
21	
22	BeginNamespace
23	
24	[C#] protected virtual void BeginNamespace();
25	[C++] protected: virtual void BeginNamespace();

		[VB] Overridable Protected Sub BeginNamespace()
	1	
	2	[JScript] protected function BeginNamespace();
	3	
	4	Description
	5	
	6	EndClass
	7	
	8	[C#] protected virtual void EndClass();
	9	[C++] protected: virtual void EndClass();
ill.	10	[VB] Overridable Protected Sub EndClass()
նուն ծում Վոյի կուս հայի կոմի հույի կույի	11	[JScript] protected function EndClass();
fun funt	12	
Just that	Ì	Description
	13	Description
	14	
if this this feather from	15	EndNamespace
die.	16	
i.	17	[C#] protected virtual void EndNamespace();
	18	[C++] protected: virtual void EndNamespace();
	19	[VB] Overridable Protected Sub EndNamespace()
	20	[JScript] protected function EndNamespace();
	21	
	22	Description
	23	
	24	GenerateMethod
	25	
		••

1	
2	[C#] protected abstract CodeMemberMethod GenerateMethod();
3	[C++] protected: virtual CodeMemberMethod* GenerateMethod() = 0;
4	[VB] MustOverride Protected Function GenerateMethod() As
5	CodeMemberMethod
6	[JScript] protected abstract function GenerateMethod(): CodeMemberMethod;
7	
8	Description
9	
10	IsBindingSupported
11	
12	[C#] protected abstract bool IsBindingSupported();
13	[C++] protected: virtual bool IsBindingSupported() = 0;
14	[VB] MustOverride Protected Function IsBindingSupported() As Boolean
15	[JScript] protected abstract function IsBindingSupported(): Boolean;
16	
17	Description
18	
19	IsOperationFlowSupported
20	
21	[C#] protected abstract bool IsOperationFlowSupported(OperationFlow flow);
22	[C++] protected: virtual bool IsOperationFlowSupported(OperationFlow flow) =
23	0;
24	[VB] MustOverride Protected Function IsOperationFlowSupported(ByVal flow
25	As OperationFlow) As Boolean

	1	[JScript] protected abstract function IsOperationFlowSupported(flow:
	2	OperationFlow): Boolean;
	3	
	4	Description
	5	
	6	OperationBindingSyntaxException
	7	
	8	[C#] public Exception OperationBindingSyntaxException(string text);
	9	[C++] public: Exception* OperationBindingSyntaxException(String* text);
ž.	10	[VB] Public Function OperationBindingSyntaxException(ByVal text As String)
	11	As Exception
11 11 11 11 11 11 11 11 11 11 11 11 11	12	[JScript] public function OperationBindingSyntaxException(text : String) :
	13	Exception;
; !	14	
	15	Description
	16	
	17	OperationSyntaxException
	18	
	19	[C#] public Exception OperationSyntaxException(string text);
	20	[C++] public: Exception* OperationSyntaxException(String* text);
	21	[VB] Public Function OperationSyntaxException(ByVal text As String) As
	22	Exception
	23	[JScript] public function OperationSyntaxException(text : String) : Exception;
	24	
	25	Description

[C#] public void UnsupportedBindingWarning(string text);[C++] public: void UnsupportedBindingWarning(String\* text);[VB] Public Sub UnsupportedBindingWarning(ByVal text As String)

[JScript] public function UnsupportedBindingWarning(text : String);

Description

# UnsupportedOperationBindingWarning

[C#] public void UnsupportedOperationBindingWarning(string text);
 [C++] public: void UnsupportedOperationBindingWarning(String\* text);
 [VB] Public Sub UnsupportedOperationBindingWarning(ByVal text As String)
 [JScript] public function UnsupportedOperationBindingWarning(text: String);

Description

## UnsupportedOperationWarning

[C#] public void UnsupportedOperationWarning(string text);
 [C++] public: void UnsupportedOperationWarning(String\* text);
 [VB] Public Sub UnsupportedOperationWarning(ByVal text As String)
 [JScript] public function UnsupportedOperationWarning(text: String);

1	
2	Description
3	
4	ProtocolReflector class (System.Web.Services.Description)
5	UnsupportedOperationWarning
6	
7	
8	Description
9	
10	ProtocolReflector
11	Example Syntax:
12	UnsupportedOperationWarning
13	
14	[C#] protected ProtocolReflector();
15	[C++] protected: ProtocolReflector();
16	[VB] Protected Sub New()
17	[JScript] protected function ProtocolReflector();
18	Binding
19	UnsupportedOperationWarning
20	
21	[C#] public Binding Binding {get;}
22	[C++] public:property Binding* get_Binding();
23	[VB] Public ReadOnly Property Binding As Binding
24	[JScript] public function get Binding() : Binding;
25	

	1	
	2	Description
	3	
	4	DefaultNamespace
	5	UnsupportedOperationWarning
	6	
	7	[C#] public string DefaultNamespace {get;}
	8	[C++] public:property String* get_DefaultNamespace();
	9	[VB] Public ReadOnly Property DefaultNamespace As String
	10	[JScript] public function get DefaultNamespace(): String;
	11	
	12	Description
	13	
ar .	14	HeaderMessages
11- 4 11 15 54 15- 27 16 16 16 16 16 16 16 16 16 16 16 16 16	15	UnsupportedOperationWarning
* H	16	
alia	17	[C#] public MessageCollection HeaderMessages {get;}
	18	[C++] public:property MessageCollection* get_HeaderMessages();
	19	[VB] Public ReadOnly Property HeaderMessages As MessageCollection
	20	[JScript] public function get HeaderMessages(): MessageCollection;
	21	
	22	Description
	23	
	24	InputMessage
	25	UnsupportedOperationWarning

```
[C#] public Message InputMessage {get;}
    [C++] public: property Message* get InputMessage();
3
    [VB] Public ReadOnly Property InputMessage As Message
    [JScript] public function get InputMessage(): Message;
6
    Description
7
8
          Method
9
          UnsupportedOperationWarning
10
11
    [C#] public LogicalMethodInfo Method {get;}
12
    [C++] public: property LogicalMethodInfo* get Method();
13
    [VB] Public ReadOnly Property Method As LogicalMethodInfo
14
    [JScript] public function get Method(): LogicalMethodInfo;
15
16
    Description
17
18
          MethodAttribute
19
          UnsupportedOperationWarning
20
21
    [C#] public WebMethodAttribute MethodAttribute {get;}
22
    [C++] public: property WebMethodAttribute* get MethodAttribute();
23
    [VB] Public ReadOnly Property MethodAttribute As WebMethodAttribute
24
    [JScript] public function get MethodAttribute(): WebMethodAttribute;
```

1	
1	
2	Description
3	
4	Methods
5	UnsupportedOperationWarning
6	
7	[C#] public LogicalMethodInfo[] Methods {get;}
8	[C++] public:property LogicalMethodInfo* get_Methods();
9	[VB] Public ReadOnly Property Methods As LogicalMethodInfo ()
10	[JScript] public function get Methods() : LogicalMethodInfo[];
11	
12	Description
13	
14	Operation
15	UnsupportedOperationWarning
16	
17	[C#] public Operation Operation {get;}
18	[C++] public:property Operation* get_Operation();
19	[VB] Public ReadOnly Property Operation As Operation
20	[JScript] public function get Operation(): Operation;
21	
22	Description
23	
24	OperationBinding
25	UnsupportedOperationWarning

```
[C#] public OperationBinding OperationBinding {get;}
    [C++] public: property OperationBinding* get OperationBinding();
3
    [VB] Public ReadOnly Property OperationBinding As OperationBinding
    [JScript] public function get OperationBinding(): OperationBinding;
5
6
    Description
7
8
           OutputMessage
9
           UnsupportedOperationWarning
10
11
    [C#] public Message OutputMessage {get;}
12
    [C++] public: property Message* get OutputMessage();
13
    [VB] Public ReadOnly Property OutputMessage As Message
14
    [JScript] public function get OutputMessage(): Message;
15
16
    Description
17
18
           Port
19
           UnsupportedOperationWarning
20
21
    [C#] public Port Port {get;}
22
    [C++] public: __property Port* get_Port();
23
    [VB] Public ReadOnly Property Port As Port
24
    [JScript] public function get Port(): Port;
```

```
Description
2
3
           PortType
           UnsupportedOperationWarning
6
    [C#] public PortType PortType {get;}
7
    [C++] public: _property PortType* get_PortType();
8
    [VB] Public ReadOnly Property PortType As PortType
9
    [JScript] public function get PortType() : PortType;
10
11
    Description
12
13
           ProtocolName
14
           UnsupportedOperationWarning
15
16
    [C#] public abstract string ProtocolName {get;}
17
    [C++] public: property virtual String* get ProtocolName() = 0;
18
    [VB] MustOverride Public ReadOnly Property ProtocolName As String
19
    [JScript] public abstract function get ProtocolName(): String;
20
21
    Description
22
23
           ReflectionImporter
24
           UnsupportedOperationWarning
25
```

1 [C#] public XmlReflectionImporter ReflectionImporter {get;} 2 [C++] public: property XmlReflectionImporter\* get ReflectionImporter(); 3 [VB] Public ReadOnly Property ReflectionImporter As XmlReflectionImporter 4 [JScript] public function get ReflectionImporter(): XmlReflectionImporter; 5 6 Description 7 8 SchemaExporter 9 **UnsupportedOperationWarning** 10 11 [C#] public XmlSchemaExporter SchemaExporter {get;} 12 [C++] public: \_property XmlSchemaExporter\* get SchemaExporter(); 13 [VB] Public ReadOnly Property SchemaExporter As XmlSchemaExporter 14 [JScript] public function get SchemaExporter(): XmlSchemaExporter; 15 16 Description 17 18 Schemas 19 **UnsupportedOperationWarning** 20 21 [C#] public XmlSchemas Schemas {get;} 22 [C++] public: property XmlSchemas\* get Schemas(); 23 [VB] Public ReadOnly Property Schemas As XmlSchemas 24 [JScript] public function get Schemas(): XmlSchemas;

1	
2	Description
3	
4	Service
5	UnsupportedOperationWarning
6	
7	[C#] public Service Service {get;}
8	[C++] public:property Service* get_Service();
9	[VB] Public ReadOnly Property Service As Service
10	[JScript] public function get Service() : Service;
11	
12	Description
13	
14	ServiceDescription
15	UnsupportedOperationWarning
16	
17	[C#] public ServiceDescription ServiceDescription {get;}
18	[C++] public:property ServiceDescription* get_ServiceDescription();
19	[VB] Public ReadOnly Property ServiceDescription As ServiceDescription
20	[JScript] public function get ServiceDescription(): ServiceDescription;
21	
22	Description
23	
24	ServiceDescriptions
25	UnsupportedOperationWarning

1	
2	[C#] public ServiceDescriptionCollection ServiceDescriptions {get;}
3	[C++] public:property ServiceDescriptionCollection*
4	get_ServiceDescriptions();
5	[VB] Public ReadOnly Property ServiceDescriptions As
6	ServiceDescriptionCollection
7	[JScript] public function get ServiceDescriptions(): ServiceDescriptionCollection;
8	
9	Description
10	
11	ServiceType
12	UnsupportedOperationWarning
13	
14	[C#] public Type ServiceType {get;}
15	[C++] public:property Type* get_ServiceType();
16	[VB] Public ReadOnly Property ServiceType As Type
17	[JScript] public function get ServiceType(): Type;
18	
19	Description
20	
21	ServiceUrl
22	UnsupportedOperationWarning
23	
24	[C#] public string ServiceUrl {get;}
25	[C++] public:property String* get_ServiceUrl();

1	[VB] Public ReadOnly Property ServiceUrl As String
2	[JScript] public function get ServiceUrl(): String;
3	
4	Description
5	
6	BeginClass
7	
8	[C#] protected virtual void BeginClass();
9	[C++] protected: virtual void BeginClass();
10	[VB] Overridable Protected Sub BeginClass()
11	[JScript] protected function BeginClass();
12	
13	Description
14	
15	EndClass
16	
17	[C#] protected virtual void EndClass();
18	[C++] protected: virtual void EndClass();
19	[VB] Overridable Protected Sub EndClass()
20	[JScript] protected function EndClass();
21	
22	Description
23	
24	GetServiceDescription
25	

1	
2	[C#] public ServiceDescription GetServiceDescription(string ns);
3	[C++] public: ServiceDescription* GetServiceDescription(String* ns);
4	[VB] Public Function GetServiceDescription(ByVal ns As String) As
5	ServiceDescription
6	[JScript] public function GetServiceDescription(ns : String) : ServiceDescription;
7	
8	Description
9	
10	ReflectMethod
11	
12	[C#] protected abstract bool ReflectMethod();
13	[C++] protected: virtual bool ReflectMethod() = 0;
14	[VB] MustOverride Protected Function ReflectMethod() As Boolean
15	[JScript] protected abstract function ReflectMethod(): Boolean;
16	
17	Description
18	
19	ReflectMethodBinding
20	
21	[C#] protected virtual string ReflectMethodBinding();
22	[C++] protected: virtual String* ReflectMethodBinding();
23	[VB] Overridable Protected Function ReflectMethodBinding() As String
24	[JScript] protected function ReflectMethodBinding(): String;
25	

	1	
	2	Description
	3	
	4	Service class (System.Web.Services.Description)
	5	ToString
	6	
	7	
	8	Description
	9	Groups together a set of related System.Web.Services.Description.Port
2000	10	instances associated with a Web Service. This class cannot be inherited.
o time suit Griff	11	Service
	12	Example Syntax:
: 	13	ToString
	14	
	15	[C#] public Service();
	16	[C++] public: Service();
	17	[VB] Public Sub New()
	18	[JScript] public function Service();
	19	Documentation
	20	Extensions
	21	ToString
	22	
	23	
	24	Description
	25	

Gets the collection of extensibility elements contained in the System.Web.Services.Description.Service instance. This property will generally only be populated with members if the 3 members of the System. Web. Services. Description. Service. Ports property lack extensibility elements. Name 6 **ToString** 8 [C#] public string Name {get; set;} [C++] public: \_\_property String\* get\_Name();public: \_\_property void 10 set\_Name(String\*); 11 [VB] Public Property Name As String 12 [JScript] public function get Name(): String; public function set Name(String); 13 14 Description 15 Gets or sets the name of the System. Web. Services. Description. Service 16 instance. 17 **Ports** 18 **ToString** 19 20 [C#] public PortCollection Ports {get;} 21 [C++] public: property PortCollection\* get Ports(); 22 [VB] Public ReadOnly Property Ports As PortCollection 23 [JScript] public function get Ports(): PortCollection;

1	
2	Description
3	Gets the collection of System.Web.Services.Description.Port instances
4	contained in the System. Web. Services. Description. Service instance.
5	ServiceDescription
6	ToString
7	
8	[C#] public ServiceDescription ServiceDescription {get;}
9	[C++] public:property ServiceDescription* get_ServiceDescription();
10	[VB] Public ReadOnly Property ServiceDescription As ServiceDescription
11	[JScript] public function get ServiceDescription(): ServiceDescription;
12	
13	Description
14	Gets the System.Web.Services.Description.ServiceDescription of which
15	the Service instance is a member.
16	ServiceCollection class (System.Web.Services.Description)
17	ToString
18	
19	
20	Description
21	Represents a collection of System.Web.Services.Description.Service
22	objects. This class cannot be inherited.
23	Count
24	InnerList
25	Item

**ToString** 

System.Web.Services.Description.Service 3 Description Gets or sets the value of a System. Web. Services. Description. Service at 5 the specified zero-based index. The zero-based index of the Service to be modified or returned. Item 8 **ToString** 9 10 [C#] public Service this[string name] {get;} 11 [C++] public: property Service\* get Item(String\* name); [VB] Public Default ReadOnly Property Item(ByVal name As String) As Service 13 [JScript] returnValue = ServiceCollectionObject.Item(name); 15 Description 16 Gets the Service instance by the value of its 17 System.Web.Services.Description.Service.Name property. The string value 18 representing the name of the System. Web. Services. Description. Service returned. List 20 Table 21 Add 22 23 [C#] public int Add(Service service);

[C++] public: int Add(Service\* service);

23

24

25

1	[VB] Public Function Add(ByVal service As Service) As Integer
2	[JScript] public function Add(service : Service) : int;
3	
4	Description
5	Adds the specified System.Web.Services.Description.Service instance to
6	the end of the ServiceCollection.
7	Return Value: Returns the zero-based index where service has been added. The
8	System.Web.Services.Description.Service instance to be added to the
9	ServiceCollection.
10	Contains
11	
12	[C#] public bool Contains(Service service);
13	[C++] public: bool Contains(Service* service);
14	[VB] Public Function Contains(ByVal service As Service) As Boolean
15	[JScript] public function Contains(service : Service) : Boolean;
16	
17	Description
18	Gets a value indicating whether the specified
19	System.Web.Services.Description.Service instance is a member of the
20	ServiceCollection .
21	Return Value: true if the specified System. Web. Services. Description. Service

service); service As Service) As Boolean ice: Service): Boolean; the specified rvice instance is a member of the tem.Web.Services.Description.Service instance is a member of the ServiceCollection; otherwise, false. A System.Web.Services.Description.Service object. CopyTo

[C#] public void CopyTo(Service[] array, int index);
[C++] public: void CopyTo(Service\* array[], int index);
[VB] Public Sub CopyTo(ByVal array() As Service, ByVal index As Integer)
[JScript] public function CopyTo(array : Service[], index : int);

Description

Copies the entire **ServiceCollection** to a one-dimensional array of type **System.Web.Services.Description.Service**, starting at the specified zero-based index of the target array. An array of type

System.Web.Services.Description.Service serving as the destination of the copy action. The zero-based index at which to start placing the copied ServiceCollection.

GetKey

[C#] protected override string GetKey(object value);

[C++] protected: String\* GetKey(Object\* value);

[VB] Overrides Protected Function GetKey(ByVal value As Object) As String [JScript] protected override function GetKey(value : Object) : String;

Description

Returns the name of the **System.Web.Services.Description.Service** associated with the value passed by reference. An object for which to return the name.

IndexOf

2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22

[C#] public int IndexOf(Service service);
[C++] public: int IndexOf(Service* service);
[VB] Public Function IndexOf(ByVal service As Service) As Integer
[JScript] public function IndexOf(service : Service) : int;

### Description

Searches for the specified **System.Web.Services.Description.Service** instance and returns the zero-based index of the first occurrence within the **ServiceCollection**.

Return Value: Returns a 32-bit signed integer. A

System.Web.Services.Description.Service object.

Insert

```
[C#] public void Insert(int index, Service service);
[C++] public: void Insert(int index, Service* service);
[VB] Public Sub Insert(ByVal index As Integer, ByVal service As Service)
[JScript] public function Insert(index : int, service : Service);
```

## Description

Adds the specified **System.Web.Services.Description.Service** instance to the **ServiceCollection** at the specified zero-based index.

If the number of items in the collection already equals the collection's capacity, the capacity is doubled by automatically reallocating the internal array before the new element is inserted. The zero-based index at which to insert the

	1	specified System.Web.Services.Description.Service. The
	2	System.Web.Services.Description.Service instance to be added to the
	3	ServiceCollection.
	4	Remove
	5	
	6	[C#] public void Remove(Service service);
	7	[C++] public: void Remove(Service* service);
	8	[VB] Public Sub Remove(ByVal service As Service)
	9	[JScript] public function Remove(service : Service);
	10	
S	11	Description
in in the second	12	Removes the first occurrence of the specified
	13	System.Web.Services.Description.Service from the ServiceCollection .
	14	This method performs a linear search; therefore, the average execution time
	15	is proportional to System.Web.Services.Description.ServiceCollection.Count.
	16	A System.Web.Services.Description.Service object.
#4 #4	17	SetParent
	18	
	19	[C#] protected override void SetParent(object value, object parent);
	20	[C++] protected: void SetParent(Object* value, Object* parent);
	21	[VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As
	22	Object)
	23	[JScript] protected override function SetParent(value : Object, parent : Object);
	24	
	25	Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Sets the parent System. Web. Services. Description. Service Description of a member of System. Web. Services. Description. Service Collection. An object, of type **System.Web.Services.Description.Service**, within the **ServiceCollection**. The object, of type System. Web. Services. Description. Service Description, to be set as the parent. ServiceDescription class (System.Web.Services.Description)

**ToString** 

#### Description

Provides a means of creating and formatting a valid WSDL (XML) document file, complete with appropriate namespaces, elements and attributes, for describing a Web Service. This class cannot be inherited.

Instances of this class can be created with either the new keyword or the static

System.Web.Services.Description.ServiceDescription.Read(System.IO.TextRe ader) method, which parses a WSDL (Web Service Description Language) file and assigns its values to appropriate members of the class.

**ToString** 

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;

25

24

	1	
	2	Description
	3	The XML namespace in which the ServiceDescription class is defined
	4	("http://schemas.xmlsoap.org/wsdl/"). This field is constant.
	5	ServiceDescription
	6	Example Syntax:
	7	ToString
	8	
	9	[C#] public ServiceDescription();
	10	[C++] public: ServiceDescription();
	11	[VB] Public Sub New()
	12	[JScript] public function ServiceDescription();
	13	Bindings
	14	ToString
	15	
a a a a a a a a a a a a a a a a a a a	16	[C#] public BindingCollection Bindings {get;}
ပေါ်န	17	[C++] public:property BindingCollection* get_Bindings();
	18	[VB] Public ReadOnly Property Bindings As BindingCollection
	19	[JScript] public function get Bindings(): BindingCollection;
	20	
	21	Description
	22	Represents the collection of System.Web.Services.Description.Binding
	23	elements that the System. Web. Services. Description. Service Description
	24	contains.
	25	Documentation

Extensions
ToString
Description
Gets the collection of extensibility elements that the
System.Web.Services.Description.ServiceDescription contains.
Imports
ToString
[C#] public ImportCollection Imports {get;}
[C++] public:property ImportCollection* get_Imports();
[VB] Public ReadOnly Property Imports As ImportCollection
[JScript] public function get Imports(): ImportCollection;
Description
Gets the collection of System. Web. Services. Description. Import elements
that the System. Web. Services. Description. Service Description contains.
Messages
ToString
[C#] public MessageCollection Messages {get;}
[C++] public:property MessageCollection* get_Messages();
[VB] Public ReadOnly Property Messages As MessageCollection
[JScript] public function get Messages(): MessageCollection;

1	
2	Description
3	The collection of System.Web.Services.Description.Message elements
4	the System. Web. Services. Description. Service Description contains.
5	Name
6	ToString
7	
8	[C#] public string Name {get; set;}
9	[C++] public:property String* get_Name();public:property void
10	set_Name(String*);
11	[VB] Public Property Name As String
12	[JScript] public function get Name(): String; public function set Name(String);
13	
14	Description
15	Gets or sets the XML Name attribute of the Descriptions tag enclosing the
16	WSDL file.
17	Returns an empty string ("") if this property value has not been assigned.
18	PortTypes
19	ToString
20	
21	[C#] public PortTypeCollection PortTypes {get;}
22	[C++] public:property PortTypeCollection* get_PortTypes();
23	[VB] Public ReadOnly Property PortTypes As PortTypeCollection
24	[JScript] public function get PortTypes(): PortTypeCollection;
25	

	5
	6
	7
	8
	9
	10
	11
	12
	13
	14
il il	15
	16
***	17
	18
	19
	20
	21

23

24

25

Description

2

3

The collection of **System.Web.Services.Description.PortType** elements the **System.Web.Services.Description.ServiceDescription** contains.

RetrievalUrl

**ToString** 

[C#] public string RetrievalUrl {get; set;}

[C++] public: \_\_property String\* get\_RetrievalUrl();public: \_\_property void set RetrievalUrl(String\*);

[VB] Public Property RetrievalUrl As String

[JScript] public function get RetrievalUrl() : String; public function set

RetrievalUrl(String);

Serializer

**ToString** 

[C#] public static XmlSerializer Serializer {get;}

[C++] public: property static XmlSerializer\* get Serializer();

[VB] Public Shared ReadOnly Property Serializer As XmlSerializer

[JScript] public static function get Serializer() : XmlSerializer; Returns the

serializer for processing web service calls. The serializer is customized according

to settings in config.web.

ServiceDescriptions

**ToString** 

1	
2	[C#] public ServiceDescriptionCollection ServiceDescriptions {get;}
3	[C++] public:property ServiceDescriptionCollection*
4	get_ServiceDescriptions();
5	[VB] Public ReadOnly Property ServiceDescriptions As
6	ServiceDescriptionCollection
7	[JScript] public function get ServiceDescriptions() : ServiceDescriptionCollection;
8	
9	Description
10	The System.Web.Services.Description.ServiceDescriptionCollection
11	instance of which the System. Web. Services. Description. Service Description is a
12	member.
13	This property is read-only.
14	Services
15	ToString
16	
17	[C#] public ServiceCollection Services {get;}
18	[C++] public:property ServiceCollection* get_Services();
19	[VB] Public ReadOnly Property Services As ServiceCollection
20	[JScript] public function get Services() : ServiceCollection;
21	
22	Description
23	Gets the collection of System.Web.Services.Description.Service instances
24	that the System. Web. Services. Description. Service Description contains.
25	TargetNamespace

1	ToString
2	
3	[C#] public string TargetNamespace {get; set;}
4	[C++] public:property String* get_TargetNamespace();public:property void
5	set_TargetNamespace(String*);
6	[VB] Public Property TargetNamespace As String
7	[JScript] public function get TargetNamespace(): String;public function set
8	TargetNamespace(String);
9	
10	Description
11	Gets or sets the XML targetNamespace attribute of the Descriptions tag
12	enclosing a WSDL file.
13	Types
14	ToString
15	
16	[C#] public Types Types {get; set;}
17	[C++] public:property Types* get_Types();public:property void
18	set_Types(Types*);
19	[VB] Public Property Types As Types
20	[JScript] public function get Types(): Types; public function set Types(Types);
21	
22	Description
23	Gets or sets the System.Web.Services.Description.Types contained by the
24	ServiceDescription instance.
25	CanRead

1	
2	[C#] p
3	[C++]
4	[VB] F
5	[JScrip
6	
7	Descri
8	
9	valid V
10	Returr
11	recogn
12	otherv
13	
14	
15	[C#] r
16	[C++]
17	[VB]
18	Servi
19	[JScri
.20	
21	Desci
22	
23	Syste
24	1
26	. II Retur

public static bool CanRead(XmlReader reader);

public: static bool CanRead(XmlReader\* reader);

Public Shared Function CanRead(ByVal reader As XmlReader) As Boolean pt] public static function CanRead(reader : XmlReader) : Boolean;

iption

Gets a value indicating whether an System.Xml.XmlReader represents a WSDL file that can be parsed.

n Value: True if the System.Xml.Serialization.XmlSerializer can nize the node on which the System.Xml.XmlReader is positioned; vise false . An System.Xml.XmlReader

Read

public static ServiceDescription Read(Stream stream);

public: static ServiceDescription\* Read(Stream\* stream);

Public Shared Function Read(ByVal stream As Stream) As

ceDescription

ipt] public static function Read(stream : Stream) : ServiceDescription;

ription

Initializes an instance of a

m.Web.Services.Description.ServiceDescription object by directly loading ML from a System.IO.Stream instance.

25 Return Value: Returns a System.Web.Services.Description.ServiceDescription

object. A System.IO.Stream object, passed by reference, that contains the bytes to be read in. 2 Read 3 [C#] public static ServiceDescription Read(string fileName); 5 [C++] public: static ServiceDescription\* Read(String\* fileName); [VB] Public Shared Function Read(ByVal fileName As String) As 7 ServiceDescription 8 [JScript] public static function Read(fileName : String) : ServiceDescription; 9 10 Description 11 Initializes an instance of a 12 System. Web. Services. Description. Service Description object by directly loading 13 the XML from the specified file. 14 Return Value: Returns a System. Web. Services. Description. Service Description 15 object. A string representing the path to the file to be read in. 16 Read 17 18 [C#] public static ServiceDescription Read(TextReader textReader); 19 [C++] public: static ServiceDescription\* Read(TextReader\* textReader); 20 [VB] Public Shared Function Read(ByVal textReader As TextReader) As 21 ServiceDescription 22 [JScript] public static function Read(textReader : TextReader) : 23 ServiceDescription; Initializes an instance of a 24 System.Web.Services.Description.ServiceDescription object by directly loading the XML.

2

1

3

5

7

8

9

10

11

13

14

15

16

17

18

19

20

21

22

23

24

25

Description

Initializes an instance of a

System.Web.Services.Description.ServiceDescription object by directly loading the XML from a System.IO.TextReader instance.

Return Value: Returns a System.Web.Services.Description.ServiceDescription object. A System.IO.TextReader object, passed by reference, that contains the text to be read in.

Read

[C#] public static ServiceDescription Read(XmlReader reader);

[C++] public: static ServiceDescription\* Read(XmlReader\* reader);

[VB] Public Shared Function Read(ByVal reader As XmlReader) As

ServiceDescription

[JScript] public static function Read(reader : XmlReader) : ServiceDescription;

Description

Initializes an instance of a

System.Web.Services.Description.ServiceDescription object by directly loading the XML from an System.Xml.XmlReader instance.

Return Value: Returns a System.Web.Services.Description.ServiceDescription object. An System.Xml.XmlReader instance, passed by reference, that contains the XML data to be read in.

Write

```
[C#] public void Write(Stream stream);
    [C++] public: void Write(Stream* stream);
3
    [VB] Public Sub Write(ByVal stream As Stream)
4
    [JScript] public function Write(stream : Stream);
5
6
    Description
7
           Writes out the System. Web. Services. Description. Service Description to
8
    the specified System.IO.Stream instance. A System.IO.Stream, passed by
9
    reference, that contains the WSDL file produced.
10
           Write
11
12
    [C#] public void Write(string fileName);
13
    [C++] public: void Write(String* fileName);
14
    [VB] Public Sub Write(ByVal fileName As String)
15
    [JScript] public function Write(fileName : String); Writes out the
16
    System.Web.Services.Description.ServiceDescription as a WSDL file.
17
18
    Description
19
           Writes out the System. Web. Services. Description. Service Description as a
20
    WSDL file to the specified path. A string representing the physical path to which
21
    the WSDL file is written.
22
           Write
23
24
    [C#] public void Write(TextWriter writer);
```

```
[C++] public: void Write(TextWriter* writer);
    [VB] Public Sub Write(ByVal writer As TextWriter)
2
    [JScript] public function Write(writer: TextWriter);
3
4
    Description
5
           Writes out the System. Web. Services. Description. Service Description to
6
    the System.IO.TextWriter instance. A System.IO.TextWriter instance.
           Write
8
9
    [C#] public void Write(XmlWriter writer);
10
    [C++] public: void Write(XmlWriter* writer);
11
    [VB] Public Sub Write(ByVal writer As XmlWriter)
12
    [JScript] public function Write(writer: XmlWriter);
13
14
    Description
15
           Writes out the System. Web. Services. Description. Service Description to
16
    the System.Xml.XmlWriter as a WSDL file. An System.Xml.XmlWriter,
17
    passed by reference, that contains the WSDL file produced.
18
           ServiceDescriptionBaseCollection class
19
    (System.Web.Services.Description)
20
           Write
21
22
23
    Description
24
25
```

Forms the basis for the strongly typed collections that are members of the **System.Web.Services.Description** namespace.

Count

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

InnerList

List

Table

Write

Description

Gets an interface that implements the association of the keys and values in the System.Web.Services.Description.ServiceDescriptionBaseCollection instance.

The default implementation returns a **System.Collections.Hashtable** to associate the keys and values. This protected method is accessible only through this class or a derived class.

**GetKey** 

[C#] protected virtual string GetKey(object value);

[C++] protected: virtual String\* GetKey(Object\* value);

[VB] Overridable Protected Function GetKey(ByVal value As Object) As String

[JScript] protected function GetKey(value : Object) : String;

Description

Returns the name of the key associated with the value passed by reference.

3

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

The default implementation of this method is intended to be overridden by a derived class to return the name of the key associated with *value*. Note that the default implementation returns null ( **Nothing** in Visual Basic). This protected method is accessible only through this class or a derived class. An object for which to return the name of the key.

OnClear

[C#] protected override void OnClear();

[C++] protected: void OnClear();

[VB] Overrides Protected Sub OnClear()

[JScript] protected override function OnClear();

Description

Clears the contents of the

## System.Web.Services.Description.ServiceDescriptionBaseCollection instance.

This method is intended to be overridden to perform additional processes when clearing the contents of a derived Collection. This protected method is accessible only through this class or a derived class.

**OnInsertComplete** 

[C#] protected override void OnInsertComplete(int index, object value);

[C++] protected: void OnInsertComplete(int index, Object\* value);

[VB] Overrides Protected Sub OnInsertComplete(ByVal index As Integer, ByVal

value As Object)

[JScript] protected override function OnInsertComplete(index : int, value :

Object);

Description

Performs additional custom processes when inserting a new element into the System.Web.Services.Description.ServiceDescriptionBaseCollection instance.

The default implementation of this method is intended to be overridden by a derived class to perform some action when the specified element is inserted. The zero-based index at which to insert the object represented by the *value* parameter. The new value of the element at the specified zero-based index.

OnRemove

[C#] protected override void OnRemove(int index, object value);

[C++] protected: void OnRemove(int index, Object\* value);

[VB] Overrides Protected Sub OnRemove(ByVal index As Integer, ByVal value As Object)

[JScript] protected override function OnRemove(index : int, value : Object);

Description

Removes an element from the

System. Web. Services. Description. Service Description Base Collection instance.

This method is intended to be overridden to perform additional processes when an element is removed from the **ServiceDescriptionBaseCollection** instance. The zero-based index at which the object represented by the *value* parameter can be found. The object to remove from the collection.

**OnSet** 

2

3

5

7

8

9

10

11

13

15 16

17

18 19

20 21

22

23 24

As Object)

[C#] protected override void OnSet(int index, object oldValue, object newValue);
[C++] protected: void OnSet(int index, Object\* oldValue, Object\* newValue);
[VB] Overrides Protected Sub OnSet(ByVal index As Integer, ByVal oldValue As Object, ByVal newValue As Object)

[JScript] protected override function OnSet(index : int, oldValue : Object,

Description

newValue: Object);

Replaces one value with another within the

System. Web. Services. Description. Service Description Base Collection instance.

The default implementation of this method is intended to be overridden to perform additional processes when setting the value of an element in the **ServiceDescriptionBaseCollection** instance. This protected method is accessible only through this class or a derived class. The zero-based index where the object represented by the *oldValue* parameter can be found. The object to replace with the object represented by the *newValue* parameter. The object that replaces the object represented by the *oldValue* parameter.

SetParent

[C#] protected virtual void SetParent(object value, object parent);

[C++] protected: virtual void SetParent(Object\* value, Object\* parent);

[VB] Overridable Protected Sub SetParent(ByVal value As Object, ByVal parent

[JScript] protected function SetParent(value : Object, parent : Object); 2 Description Sets the Parent property of the System. Web. Services. Description. Service Description Base Collection instance. The default implementation of this method is intended to be overridden by 6 a derived class to set a parent of an appropriate class. Note also that the default 7 implementation performs no action. This protected method is accessible only 8 through this class or a derived class. The object for which to set the parent object. The object to set as the parent. 10 ServiceDescriptionCollection class (System.Web.Services.Description) 11 **ToString** 12 13 14 Description 15 Represents a collection of 16 System.Web.Services.Description.ServiceDescription instances. This class 17 cannot be inherited. 18 ServiceDescriptionCollection 19 Example Syntax: 20 **ToString** 21 22 [C#] public ServiceDescriptionCollection(); 23 [C++] public: ServiceDescriptionCollection(); 24 [VB] Public Sub New()

1	[JScript] public function ServiceDescriptionCollection();
2	
3	Description
4	Initializes a new instance of the
5	System.Web.Services.Description.ServiceDescriptionCollection class.
6	Count
7	InnerList
8	Item
9	ToString
10	System.Web.Services.Description.ServiceDescription
11	
12	Description
13	Gets or sets the value of a
14	System.Web.Services.Description.ServiceDescription at the specified zero-
15	based index. The zero-based index of the
16	System.Web.Services.Description.ServiceDescription whose value is modified
17	or returned.
18	Item
19	ToString
20	
21	[C#] public ServiceDescription this[string ns] {get;}
22	[C++] public:property ServiceDescription* get_Item(String* ns);
23	[VB] Public Default ReadOnly Property Item(ByVal ns As String) As
24	ServiceDescription
25	[JScript] returnValue = ServiceDescriptionCollectionObject.Item(ns);

Aller come have a

Description

3

7

8

9

10

11

12

13

14

15

16

17

18

20

21

22

23

24

25

Gets a **System.Web.Services.Description.ServiceDescription** specified by its **System.Web.Services.Description.ServiceDescription.TargetNamespace** property. The namespace of the

System. Web. Services. Description. Service Description to be returned.

List

Table

Add

[C#] public int Add(ServiceDescription serviceDescription);

[C++] public: int Add(ServiceDescription\* serviceDescription);

[VB] Public Function Add(ByVal serviceDescription As ServiceDescription) As Integer

[JScript] public function Add(serviceDescription : ServiceDescription) : int;

Description

 ${\bf Adds\ the\ specified\ System. Web. Services. Description. Service Description}$  to the end of the Service Description Collection .

Return Value: Returns the zero-based index where the specified

System. Web. Services. Description. Service Description instance has been added.

The **System.Web.Services.Description.ServiceDescription** to be added to the collection.

Contains

1	
2	[C#] public bool Contains(ServiceDescription serviceDescription);
3	[C++] public: bool Contains(ServiceDescription* serviceDescription);
4	[VB] Public Function Contains(ByVal serviceDescription As ServiceDescription)
5	As Boolean
6	[JScript] public function Contains(serviceDescription : ServiceDescription) :
7	Boolean;
8	
9	Description
10	Gets a value indicating whether the specified
11	System.Web.Services.Description.ServiceDescription instance is a member of
12	the collection.
13	Return Value: true if the specified ServiceDescription instance is a member of the
14	ServiceDescriptionCollection; otherwise false. A
15	System.Web.Services.Description.ServiceDescription object.
16	СоруТо
17	
18	[C#] public void CopyTo(ServiceDescription[] array, int index);
19	[C++] public: void CopyTo(ServiceDescription* array[], int index);
20	[VB] Public Sub CopyTo(ByVal array() As ServiceDescription, ByVal index As
21	Integer)
22	[JScript] public function CopyTo(array: ServiceDescription[], index: int);
23	
24	Description
25	

2

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Copies the entire ServiceDescriptionCollection to a one-dimensional array of type System. Web. Services. Description. Service Description, starting at the specified zero-based index of the target array. An array of type System. Web. Services. Description. Service Description serving as the destination of the copy action. The zero-based index at which to start placing the copied collection. GetBinding

[C#] public Binding GetBinding(XmlQualifiedName name);

[C++] public: Binding\* GetBinding(XmlQualifiedName\* name);

[VB] Public Function GetBinding(ByVal name As XmlQualifiedName) As Binding

[JScript] public function GetBinding(name : XmlQualifiedName) : Binding;

## Description

Searches the ServiceDescriptionCollection and returns the System. Web. Services. Description. Binding instance with the specified name that is a member of one of the System. Web. Services. Description. Service Description instances contained in the collection.

Return Value: A Binding object. The System.Xml.XmlQualifiedName, passed by reference, whose **Name** property is shared by the

System.Web.Services.Description.Binding returned.

GetKey

[C#] protected override string GetKey(object value);

1	[C++] protected: String* GetKey(Object* value);
2	[VB] Overrides Protected Function GetKey(ByVal value As Object) As String
3	[JScript] protected override function GetKey(value : Object) : String;
4	
5	Description
6	Returns the
7	System.Web.Services.Description.ServiceDescription.TargetNamespace
8	property of the System. Web. Services. Description. Service Description associated
9	with the value passed by reference. The object for which to return the
10	TargetNamespace.
11	GetMessage
12	
13	[C#] public Message GetMessage(XmlQualifiedName name);
14	[C++] public: Message* GetMessage(XmlQualifiedName* name);
15	[VB] Public Function GetMessage(ByVal name As XmlQualifiedName) As
16	Message
17	[JScript] public function GetMessage(name : XmlQualifiedName) : Message;
18	
19	Description
20	Searches the ServiceDescriptionCollection and returns the
21	System.Web.Services.Description.Message instance with the specified name that
22	is a member of one of the System. Web. Services. Description. Service Description
23	instances contained in the collection.
24	Return Value: A System.Web.Services.Description.Message object. The
25	

11	
i	System.Xml.XmlQualifiedName, passed by reference, whose Name property is
2	shared by the System. Web. Services. Description. Message returned.
3	GetPortType
4	
5	[C#] public PortType GetPortType(XmlQualifiedName name);
6	[C++] public: PortType* GetPortType(XmlQualifiedName* name);
7	[VB] Public Function GetPortType(ByVal name As XmlQualifiedName) As
8	PortType
9	[JScript] public function GetPortType(name : XmlQualifiedName) : PortType;
10	
11	Description
12	Searches the ServiceDescriptionCollection and returns the
13	System.Web.Services.Description.PortType instance with the specified name
14	that is a member of one of the
15	System.Web.Services.Description.ServiceDescription instances contained in the
16	collection.
17	Return Value: A System.Web.Services.Description.PortType object. The
18	System.Xml.XmlQualifiedName, passed by reference, whose Name property is
19	shared by the System. Web. Services. Description. Port Type returned.
20	GetService
21	
22	[C#] public Service GetService(XmlQualifiedName name);
23	[C++] public: Service* GetService(XmlQualifiedName* name);
24	[VB] Public Function GetService(ByVal name As XmlQualifiedName) As Service
25	[JScript] public function GetService(name : XmlQualifiedName) : Service;

Tion.	
Tirell Healt	
i dinit	
į	
Ĭ	

$\boldsymbol{r}$	esci	. • .		
,,,	occi	nn	TIO	m
$\boldsymbol{\mathcal{L}}$		$\iota \nu$	$\iota\iota\upsilon$	,,

3

5

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Searches the ServiceDescriptionCollection and returns the System.Web.Services.Description.Service instance with the specified name that is a member of one of the System.Web.Services.Description.ServiceDescription instances contained in the collection.

Return Value: A System.Web.Services.Description.Service object. The System.Xml.XmlQualifiedName, passed by reference, whose Name property is shared by the System.Web.Services.Description.Service returned.

IndexOf

[C#] public int IndexOf(ServiceDescription serviceDescription);

[C++] public: int IndexOf(ServiceDescription\* serviceDescription);

[VB] Public Function IndexOf(ByVal serviceDescription As ServiceDescription)

As Integer

[JScript] public function IndexOf(serviceDescription : ServiceDescription) : int;

Description

Searches for the specified

**System.Web.Services.Description.ServiceDescription** instance and returns the zero-based index of the first occurrence within the **ServiceDescriptionCollection** 

Return Value: Returns a 32-bit signed integer. A

System. Web. Services. Description. Service Description object.

Insert

1	
2	[C#] public void Insert(int index, ServiceDescription serviceDescription);
3	[C++] public: void Insert(int index, ServiceDescription* serviceDescription);
4	[VB] Public Sub Insert(ByVal index As Integer, ByVal serviceDescription As
5	ServiceDescription)
6	[JScript] public function Insert(index : int, serviceDescription :
7	ServiceDescription);
8	
9	Description
10	Adds the specified System. Web. Services. Description. Service Description
11	instance to the ServiceDescriptionCollection at the specified index. The zero-
12	based index at which to insert the specified
13	System.Web.Services.Description.ServiceDescription. The
14	System.Web.Services.Description.ServiceDescription to add to the collection.
15	Remove
16	
17	[C#] public void Remove(ServiceDescription serviceDescription);
18	[C++] public: void Remove(ServiceDescription* serviceDescription);
19	[VB] Public Sub Remove(ByVal serviceDescription As ServiceDescription)
20	[JScript] public function Remove(serviceDescription : ServiceDescription);
21	
22	Description
23	Removes the first occurrence of a
24	System.Web.Services.Description.ServiceDescription instance specified by the
25	$service Description\ parameter\ from\ the\ Service Description\ Collection\ .$

This method performs a linear search; therefore, the average execution time is proportional to **Count** . The

**System.Web.Services.Description.ServiceDescription** to be removed from the collection.

ServiceDescriptionFormatExtension class
(System.Web.Services.Description)

**ToString** 

## Description

Represents an extensibility element added to a Web Service.

In a derived class, the ServiceDescriptionFormatExtension class allows users to define extensibility elements in addition to those defined in the WSDL specification. Note that extensibility elements can be added to a ServiceDescription at any of several levels. Thus multiple classes derived from the System.Web.Services.Description.DocumentableItem class (including the System.Web.Services.Description.ServiceDescription class) have an Extensions property, which returns a System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect ion .

ServiceDescriptionFormatExtension

Example Syntax:

**ToString** 

[C#] protected ServiceDescriptionFormatExtension();

```
[C++] protected: ServiceDescriptionFormatExtension();
    [VB] Protected Sub New()
 2
    [JScript] protected function ServiceDescriptionFormatExtension();
           Handled
           ToString
 5
 6
    [C#] public bool Handled {get; set;}
 7
    [C++] public: property bool get Handled(); public: property void
    set_Handled(bool);
 9
    [VB] Public Property Handled As Boolean
10
    [JScript] public function get Handled(): Boolean; public function set
11
    Handled(Boolean);
12
13
    Description
14
           Gets or sets a value indicating whether the
15
    System.Web.Services.Description.ServiceDescriptionFormatExtension is
16
    handled by the action to which it refers.
17
           Parent
18
           ToString
19
20
    [C#] public object Parent {get;}
21
    [C++] public: property Object* get Parent();
22
    [VB] Public ReadOnly Property Parent As Object
23
    [JScript] public function get Parent(): Object;
24
25
```

25

Description Gets the parent object of the 3  $System. Web. Services. Description. Service Description Format Extension \ .$ When overridden in a derived class, the class returned by this property will 5 be a derived class rather than the base **System.Object** class. Required 7 **ToString** 9 [C#] public bool Required {get; set;} 10 [C++] public: property bool get\_Required();public: property void 11 set Required(bool); 12 [VB] Public Property Required As Boolean 13 [JScript] public function get Required(): Boolean; public function set 14 Required(Boolean); 15 16 Description 17 Gets or sets a value indicating whether the 18 System.Web.Services.Description.ServiceDescriptionFormatExtension is 19 necessary for the action to which it refers. 20 ServiceDescriptionFormatExtensionCollection class 21 (System.Web.Services.Description) 22 **ToString** 23

4.C
To the state of th
طاسم

1	
2	
3	Description
4	Represents the collection of extensibility elements used by the Web
5	Service. This class cannot be inherited.
6	ServiceDescriptionFormatExtensionCollection
7	Example Syntax:
8	ToString
9	
10	[C#] public ServiceDescriptionFormatExtensionCollection(object parent);
11	[C++] public: ServiceDescriptionFormatExtensionCollection(Object* parent);
12	[VB] Public Sub New(ByVal parent As Object)
13	[JScript] public function ServiceDescriptionFormatExtensionCollection(parent:
14	Object);
15	
16	Description
17	Initializes a new instance of the
18	System. We b. Services. Description. Service Description Format Extension Collect and the property of the pr
19	ion class. The System.Web.Services.Description.ServiceDescription of which
20	this collection is a member.
21	Count
22	InnerList
23	Item

ToString

1	
2	
3	Description
4	Gets or sets the value of a member of the
5	System. We b. Services. Description. Service Description Format Extension Collect
6	ion . The zero-based index of the member in question.
7	List
8	Table
9	Add
10	
11	[C#] public int Add(object extension);
12	[C++] public: int Add(Object* extension);
13	[VB] Public Function Add(ByVal extension As Object) As Integer
14	[JScript] public function Add(extension : Object) : int;
15	
16	Description
17	Adds extension to the
18	System. We b. Services. Description. Service Description Format Extension Collect
19	ion .
20	Return Value: Returns the index value of the member added. The
21	System.Web.Services.Description.ServiceDescriptionFormatExtension, passed
22	by reference, to be added to the
23	System. We b. Services. Description. Service Description Format Extension Collect
24	ion.
25	Contains

1	
2	[C#] public bool Contains(object extension);
3	[C++] public: bool Contains(Object* extension);
4	[VB] Public Function Contains(ByVal extension As Object) As Boolean
5	[JScript] public function Contains(extension : Object) : Boolean;
6	
7	Description
8	Gets a value indicating whether extension is a member of the
9	System. We b. Services. Description. Service Description Format Extension Collect
10	ion .
11	Return Value: true if the object is a member of the collection; otherwise, false.
12	The object that is to be checked whether it is a member of the
13	System. We b. Services. Description. Service Description Format Extension Collect
14	ion.
15	СоруТо
16	
17	[C#] public void CopyTo(object[] array, int index);
18	[C++] public: void CopyTo(Object* arraygc[], int index);
19	[VB] Public Sub CopyTo(ByVal array() As Object, ByVal index As Integer)
20	[JScript] public function CopyTo(array : Object[], index : int);
21	
22	Description
23	Copies the elements in the
24	System. We b. Services. Description. Service Description Format Extension Collect
25	

ion into array, starting at index. The destination of the copy action The zero-based index at which to place the first copied element

Find

[C#] public object Find(Type type);

[C++] public: Object\* Find(Type\* type);

[VB] Public Function Find(ByVal type As Type) As Object

[JScript] public function Find(type: Type): Object; This method searches the

System. Web. Services. Description. Service Description Format Extension Collect

ion and returns the first member of the collection specified by the parameter passed in.

Description

This method searches the

System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect ion and returns the first instance of *type* that it discovers.

Return Value: Returns an object of the specified **System.Type** if its search was successful; **null** otherwise.

Note that this method searches the collection in index order, and returns only the *type* with the lowest-numbered index. A **System.Type** for which to search the collection.

Find

23

24

[C#] public XmlElement Find(string name, string ns);

[C++] public: XmlElement\* Find(String\* name, String\* ns);

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
ij
12:25
14
in the second
sab
:::::: :::::::::::::::::::::::::::::::
ndi.

1	[VB] Public Function Find(ByVal name As String, ByVal ns As String) As
2	XmlElement
3	[JScript] public function Find(name : String, ns : String) : XmlElement;
4	
5	Description
6	This method searches the
7	System. We b. Services. Description. Service Description Format Extension Collect the contract of the contra
8	ion for a member with name name and namespace URI ns.
9	Return Value: Returns an System.Xml.XmlElement if the search is successful;
10	otherwise null.
11	Note that this method searches the collection in index order, and will return
12	the first System.Xml.XmlElement that meets the criteria of the two parameters. A
13	string representing the name of the System.Xml.XmlElement to be found. A
14	string representing the XML namespace URI of the System.Xml.XmlElement to
15	be found.
16	FindAll
17	
18	[C#] public object[] FindAll(Type type);
19	[C++] public: Object* FindAll(Type* type)gc[];
20	[VB] Public Function FindAll(ByVal type As Type) As Object()
21	[JScript] public function FindAll(type: Type): Object[]; This method searches the
22	System. We b. Services. Description. Service Description Format Extension Collect Property of the property o
23	ion for all members of the collection specified by the parameter passed in.
24	
25	Description

24

25

1

System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect 2 ion and returns an array of object instances of System. Typetype that it discovers. 3 Return Value: Returns an array of System. Object instances. The array returned will be empty if the search is unsuccessful. A 5 System. Type for which to search the collection. **FindAll** 7 8 [C#] public XmlElement[] FindAll(string name, string ns); 9 [C++] public: XmlElement\* FindAll(String\* name, String\* ns) []; 10 [VB] Public Function FindAll(ByVal name As String, ByVal ns As String) As 11 XmlElement() 12 [JScript] public function FindAll(name : String, ns : String) : XmlElement[]; 13 14 Description 15 This method searches the 16 System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect 17 ion for members with name name and namespace URI ns. 18 Return Value: Returns an array of System.Xml.XmlElement objects. 19 The array returned will be empty if the search is unsuccessful. A string 20 representing the name attribute of the System.Xml.XmlElement objects to be 21 found. A string representing the XML namespace URI attribute of the 22

System.Xml.XmlElement objects to be found.

IndexOf

This method searches the

(ee@haves ык 509-324-9256 658 MSI-863US.APP

1	
2	[C#] public int IndexOf(object extension);
3	[C++] public: int IndexOf(Object* extension);
4	[VB] Public Function IndexOf(ByVal extension As Object) As Integer
5	[JScript] public function IndexOf(extension : Object) : int;
6	
7	Description
8	Gets the zero-based index value of extension, the specified member of the
9	System. We b. Services. Description. Service Description Format Extension Collect
10	ion .
11	Return Value: Returns a 32-bit signed integer. The object for which to return the
12	index value.
13	Insert
14	
15	[C#] public void Insert(int index, object extension);
16	[C++] public: void Insert(int index, Object* extension);
17	[VB] Public Sub Insert(ByVal index As Integer, ByVal extension As Object)
18	[JScript] public function Insert(index : int, extension : Object);
19	
20	Description
21	Adds extension to the
22	System. We b. Services. Description. Service Description Format Extension Collect
23	ion at the specified index. The zero-based index at which to add the new member.
24	The object to add to the

1	System. We b. Services. Description. Service Description Format Extension Collect
2	ion.
3	IsHandled
4	
5	[C#] public bool IsHandled(object item);
6	[C++] public: bool IsHandled(Object* item);
7	[VB] Public Function IsHandled(ByVal item As Object) As Boolean
8	[JScript] public function IsHandled(item: Object): Boolean;
9	
10	Description
11	Gets a value indicating whether <i>item</i> is handled.
12	Return Value: true if item is handled; otherwise false. An object, either of type
13	System.Xml.XmlElement or
14	System. Web. Services. Description. Service Description Format Extension.
15	IsRequired
16	
17	[C#] public bool IsRequired(object item);
18	[C++] public: bool IsRequired(Object* item);
19	[VB] Public Function IsRequired(ByVal item As Object) As Boolean
20	[JScript] public function IsRequired(item : Object) : Boolean;
21	
22	Description
23	Gets a value indicating whether item is required.
24	Return Value: true if item is required; otherwise false. An object, either of type
25	

1 }	System.Xml.XmlElement or
2	System.Web.Services.Description.ServiceDescriptionFormatExtension.
3	OnValidate
4	
5	[C#] protected override void OnValidate(object value);
6	[C++] protected: void OnValidate(Object* value);
7	[VB] Overrides Protected Sub OnValidate(ByVal value As Object)
8	[JScript] protected override function OnValidate(value : Object);
9	
10	Description
11	Performs a check on the <b>System.Type</b> of <i>value</i> when validating it.
12	This method overrides the base method in order to determine that <i>value</i> is
13	one of the two valid classes, System.Xml.XmlElement or
14	System.Web.Services.Description.ServiceDescriptionFormatExtension . The
15	object to be validated.
16	Remove
17	
18	[C#] public void Remove(object extension);
19	[C++] public: void Remove(Object* extension);
20	[VB] Public Sub Remove(ByVal extension As Object)
21	[JScript] public function Remove(extension : Object);
22	
23	Description
24	Removes the first occurrence of extension from the
25	System. We b. Services. Description. Service Description Format Extension Collect

ion. The object to be removed from the System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect ion. 3 SetParent 5 [C#] protected override void SetParent(object value, object parent); [C++] protected: void SetParent(Object\* value, Object\* parent); [VB] Overrides Protected Sub SetParent(ByVal value As Object, ByVal parent As Object) [JScript] protected override function SetParent(value : Object, parent : Object); 10 11 Description 12 Sets the parent object of a member of the 13 System.Web.Services.Description.ServiceDescriptionFormatExtensionCollect 14 ion. 15 This method overrides the base method in order to determine that value is 16 of type System. Web. Services. Description. Service Description Format Extension 17 . A child object for which to set the parent. The object to be set as parent. 18 ServiceDescriptionImporter class (System.Web.Services.Description) 19 **ToString** 20 21 22 Description 23 This class exposes methods for generating client proxy classes for Web 24 Services.

	1	ServiceDescriptionImporter
	2	Example Syntax:
	3	ToString
	4	
	5	[C#] public ServiceDescriptionImporter();
	6	[C++] public: ServiceDescriptionImporter();
	7	[VB] Public Sub New()
	8	[JScript] public function ServiceDescriptionImporter();
	9	
in the second	10	Description
	11	Initializes a new instance of the
	12	System.Web.Services.Description.ServiceDescriptionImporter class.
	13	ProtocolName
	14	ToString
	15	
	16	[C#] public string ProtocolName {get; set;}
	17	[C++] public:property String* get_ProtocolName();public:property void
	18	set_ProtocolName(String*);
	19	[VB] Public Property ProtocolName As String
	20	[JScript] public function get ProtocolName(): String;public function set
	21	ProtocolName(String);
	22	
	23	Description
	24	Gets or sets the name of the protocol to be imported.
	25	Schemas

	1	ToString
	2	
	3	[C#] public XmlSchemas Schemas {get;}
	4	[C++] public:property XmlSchemas* get_Schemas();
	5	[VB] Public ReadOnly Property Schemas As XmlSchemas
	6	[JScript] public function get Schemas(): XmlSchemas;
	7	
	8	Description
had had but had had but he had had	9	Gets the System.Xml.Serialization.XmlSchemas Collection contained in
	10	the Web Service to be imported. This property is read-only.
	11	ServiceDescriptions
	12	ToString
Ting Ting	13	
***************************************	14	[C#] public ServiceDescriptionCollection ServiceDescriptions {get;}
	15	[C++] public:property ServiceDescriptionCollection*
	16	get_ServiceDescriptions();
	17	[VB] Public ReadOnly Property ServiceDescriptions As
	18	ServiceDescriptionCollection
	19	[JScript] public function get ServiceDescriptions(): ServiceDescriptionCollection;
	20	
	21	Description
	22	Gets the System. Web. Services. Description. Service Description Collection
	23	being imported by the ServiceDescriptionImporter.
	24	As with any collection that forms a read-only property, members can be
	25	added to the collection, removed from the collection, or modified using the

methods exposed by the collection. However, it is recommended that the developer use the 2 System.Web.Services.Description.ServiceDescriptionImporter.AddServiceDe 3 scription(System.Web.Services.Description.ServiceDescription,System.String, System.String) method to add members to this collection. 5 Style **ToString** 7 8 [C#] public ServiceDescriptionImportStyle Style {get; set;} 9 [C++] public: property ServiceDescriptionImportStyle get Style();public: 10 property void set Style(ServiceDescriptionImportStyle); 11 [VB] Public Property Style As ServiceDescriptionImportStyle 12 [JScript] public function get Style(): ServiceDescriptionImportStyle; public 13 function set Style(ServiceDescriptionImportStyle); 14 15 Description 16 Gets or sets the 17 System.Web.Services.Description.ServiceDescriptionImportStyle associated 18 with the System. Web. Services. Description. Service Description Importer 19 instance. 20 AddServiceDescription 21 22 [C#] public void AddServiceDescription(ServiceDescription, 23 string appSettingUrlKey, string appSettingBaseUrl); 24 [C++] public: void AddServiceDescription(ServiceDescription\*

2		
ŝ		
Many many		
1		

serviceDescription, String\* appSettingUrlKey, String\* appSettingBaseUrl);
[VB] Public Sub AddServiceDescription(ByVal serviceDescription As
ServiceDescription, ByVal appSettingUrlKey As String, ByVal
appSettingBaseUrl As String)

[JScript] public function AddServiceDescription(serviceDescription: ServiceDescription, appSettingUrlKey: String, appSettingBaseUrl: String);

Description

Adds the specified **System.Web.Services.Description.ServiceDescription** to the

System.Web.Services.Description.ServiceDescriptionImporter.ServiceDescriptions collection. It also sets the

System.Web.Services.Description.ServiceDescription.AppSettingUrlKey and System.Web.Services.Description.ServiceDescription.AppSettingBaseUrl properties of the System.Web.Services.Description.ServiceDescription to be added.

The two string parameters, appSettingUrlKey and appSettingBaseUrl, specify how the Url property of the Web Service proxy to be generated from the imported serviceDescription should be constructed. The System.Web.Services.Description.ServiceDescription instance to add to the collection Sets the initial value of the Url property of the proxy class to be generated from the instance represented by the serviceDescription parameter. Specifies that it should be generated from the web.config file's section. Sets the initial value of the Url property of the proxy class to be generated from the instance represented by the serviceDescription parameter. Specifies that it should

be constructed from a combination of the value of this parameter and the URL specified by the location attribute in the WSDL document.

Import

[C#] public ServiceDescriptionImportWarnings Import(CodeNamespace codeNamespace, CodeCompileUnit codeCompileUnit);

[C++] public: ServiceDescriptionImportWarnings Import(CodeNamespace\* codeNamespace, CodeCompileUnit\* codeCompileUnit);

[VB] Public Function Import(ByVal codeNamespace As CodeNamespace, ByVal codeCompileUnit As CodeCompileUnit) As ServiceDescriptionImportWarnings

[JScript] public function Import(codeNamespace: CodeNamespace, codeCompileUnit: CodeCompileUnit): ServiceDescriptionImportWarnings;

## Description

Imports a **System.Web.Services.Description.ServiceDescription** from the specified namespace, and generates code for client proxy classes.

Return Value: One of the

System.Web.Services.Description.ServiceDescriptionImportWarnings values.

This method actually does the work of importing

System. Web. Services. Description. Service Descriptions and

System.Xml.Schema.XmlSchemas . A System.CodeDom.CodeNamespace

instance that determines the namespace of the

System. Web. Services. Description. Service Description to be imported. A

System.CodeDom.CodeCompileUnit instance that will add references to the

appropriate assemblies.

1	ServiceDescriptionImportStyle enumeration
2	(System.Web.Services.Description)
3	ToString
4	
5	
6	Description
7	Specifies whether the Import is made to the server or the client machine.
8	ToString
9	
10	[C#] public const ServiceDescriptionImportStyle Client;
11	[C++] public: const ServiceDescriptionImportStyle Client;
12	[VB] Public Const Client As ServiceDescriptionImportStyle
13	[JScript] public var Client : ServiceDescriptionImportStyle;
14	
15	Description
16	Specifies that the Import should be made to the client machine.
17	ToString
18	
19	[C#] public const ServiceDescriptionImportStyle Server;
20	[C++] public: const ServiceDescriptionImportStyle Server;
21	[VB] Public Const Server As ServiceDescriptionImportStyle
22	[JScript] public var Server : ServiceDescriptionImportStyle;
23	
24	Description
25	Specifies that the Import should be made to the server.

ServiceDescriptionImportWarnings enumeration

(System.Web.Services.Description)

ToString

Description

Specifies the type of warnings produced by

ToString

[C#] public const ServiceDescriptionImportWarnings NoCodeGenerated;
 [C++] public: const ServiceDescriptionImportWarnings NoCodeGenerated;
 [VB] Public Const NoCodeGenerated As ServiceDescriptionImportWarnings
 [JScript] public var NoCodeGenerated : ServiceDescriptionImportWarnings;

System.Web.Services.Description.ServiceDescriptionImporter.Import(System

.CodeDom.CodeNamespace,System.CodeDom.CodeCompileUnit) .

Description

Specifies that no proxy class was generated by the 
System.Web.Services.Description.ServiceDescriptionImporter.Import(System .CodeDom.CodeNamespace,System.CodeDom.CodeCompileUnit) method.

**ToString** 

[C#] public const ServiceDescriptionImportWarnings NoMethodsGenerated;[C++] public: const ServiceDescriptionImportWarnings NoMethodsGenerated;[VB] Public Const NoMethodsGenerated As ServiceDescriptionImportWarnings

[JScript] public var NoMethodsGenerated : ServiceDescriptionImportWarnings; 2 Description Specifies that the proxy class generated by the System.Web.Services.Description.ServiceDescriptionImporter.Import(System .CodeDom.CodeNamespace,System.CodeDom.CodeCompileUnit) method includes no methods. **ToString** 8 9 [C#] public const ServiceDescriptionImportWarnings OptionalExtensionsIgnored; 10 [C++] public: const ServiceDescriptionImportWarnings 11 OptionalExtensionsIgnored; 12 [VB] Public Const OptionalExtensionsIgnored As 13 ServiceDescriptionImportWarnings 14 [JScript] public var OptionalExtensionsIgnored: 15 ServiceDescriptionImportWarnings; 16 17 Description 18 Specifies that at least one optional 19 System.Web.Services.Description.ServiceDescriptionFormatExtension for the 20 System. Web. Services. Description. Service Description to be imported has been 21 ignored. 22 **ToString** 23 24 [C#] public const ServiceDescriptionImportWarnings RequiredExtensionsIgnored;

1	[C++] public: const ServiceDescriptionImportWarnings
2	RequiredExtensionsIgnored;
3	[VB] Public Const RequiredExtensionsIgnored As
4	ServiceDescriptionImportWarnings
5	[JScript] public var RequiredExtensionsIgnored :
6	ServiceDescriptionImportWarnings;
7	
8	Description
9	Specifies that at least one necessary
10	System.Web.Services.Description.ServiceDescriptionFormatExtension for the
11	System.Web.Services.Description.ServiceDescription to be imported has been
12	ignored.
13	ToString
14	
15	[C#] public const ServiceDescriptionImportWarnings
16	UnsupportedBindingsIgnored;
17	[C++] public: const ServiceDescriptionImportWarnings
18	UnsupportedBindingsIgnored;
19	[VB] Public Const UnsupportedBindingsIgnored As
20	ServiceDescriptionImportWarnings
21	[JScript] public var UnsupportedBindingsIgnored:
22	ServiceDescriptionImportWarnings;
23	
24	Description
25	

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Specifies that at least one System. Web. Services. Description. Binding for the System. Web. Services. Description. Service Description to be imported is of an unsupported type and has been ignored. **ToString** [C#] public const ServiceDescriptionImportWarnings UnsupportedOperationsIgnored; [C++] public: const ServiceDescriptionImportWarnings UnsupportedOperationsIgnored; [VB] Public Const UnsupportedOperationsIgnored As **ServiceDescriptionImportWarnings** [JScript] public var UnsupportedOperationsIgnored: ServiceDescriptionImportWarnings; Description Specifies that at least one System. Web. Services. Description. Operation for the System. Web. Services. Description. Service Description to be imported is of an unsupported type and has been ignored. ServiceDescriptionReflector class (System.Web.Services.Description) **ToString** Description

ServiceDescriptionReflector

1	Example Syntax:
2	ToString
3	
4	[C#] public ServiceDescriptionReflector();
5	[C++] public: ServiceDescriptionReflector();
6	[VB] Public Sub New()
7	[JScript] public function ServiceDescriptionReflector();
8	
9	Description
10	
11	Schemas
12	ToString
13	
14	[C#] public XmlSchemas Schemas {get;}
15	[C++] public:property XmlSchemas* get_Schemas();
16	[VB] Public ReadOnly Property Schemas As XmlSchemas
17	[JScript] public function get Schemas() : XmlSchemas;
18	
19	Description
20	
21	ServiceDescriptions
22	ToString
23	
24	[C#] public ServiceDescriptionCollection ServiceDescriptions {get;}
25	[C++] public: property ServiceDescriptionCollection*

1	<pre>get_ServiceDescriptions();</pre>
2	[VB] Public ReadOnly Property ServiceDescriptions As
3	ServiceDescriptionCollection
4	[JScript] public function get ServiceDescriptions(): ServiceDescriptionCollection;
5	
6	Description
7	
8	Reflect
9	
10	[C#] public void Reflect(Type type, string url);
11	[C++] public: void Reflect(Type* type, String* url);
12	[VB] Public Sub Reflect(ByVal type As Type, ByVal url As String)
13	[JScript] public function Reflect(type : Type, url : String);
14	
15	Description
16	
17	SoapAddressBinding class (System.Web.Services.Description)
18	ToString
19	
20	
21	Description
22	Represents an extensibility element added to a
23	System.Web.Services.Description.Port within a Web Service. This class cannot
24	be inherited.
25	SoapAddressBinding

1	Example Syntax:
2	ToString
3	
4	[C#] public SoapAddressBinding();
5	[C++] public: SoapAddressBinding();
6	[VB] Public Sub New()
7	[JScript] public function SoapAddressBinding();
8	Handled
9	Location
10	ToString
11	
12	
13	Description
14	Gets or sets a value representing the URI for the
15	System.Web.Services.Description.Port to which the SoapAddressBinding
16	applies.
17	Parent
18	Required
19	SoapBinding class (System.Web.Services.Description)
20	ToString
21	
22	
23	Description
24	Represents an extension added to a
25	System.Web.Services.Description.Binding within a Web Service. It specifies

that the data transmission will use SOAP (Simple Object Access Protocol). This
class cannot be inherited.
ToString
[C#] public const string HttpTransport;
[C++] public: const String* HttpTransport;
[VB] Public Const HttpTransport As String
[JScript] public var HttpTransport : String;
Description
ToString
[C#] public const string Namespace;
[C++] public: const String* Namespace;
[VB] Public Const Namespace As String
[JScript] public var Namespace : String;
Description
Gets the URI (http://schemas.xmlsoap.org/wsdl/soap/) for the XML
namespace of the SoapBinding class. This field is constant.
SoapBinding
Example Syntax:
ToString
1

```
[C#] public SoapBinding();
    [C++] public: SoapBinding();
    [VB] Public Sub New()
    [JScript] public function SoapBinding();
           Handled
6
           Parent
7
           Required
8
           Style
9
           ToString
10
11
12
    Description
13
           Specifies the type of SOAP binding used by the SoapBinding instance.
14
           Transport
15
           ToString
16
17
    [C#] public string Transport {get; set;}
18
     [C++] public: __property String* get_Transport();public: __property void
19
     set Transport(String*);
20
     [VB] Public Property Transport As String
21
     [JScript] public function get Transport(): String; public function set
22
     Transport(String);
23
24
     Description
```

Gets or sets the URI for the specification for HTTP transmission of SOAP data.

In general usage, this required property value will be the same as that of the **System.Web.Services.Description.SoapBinding.HttpTransport** field. However, this property value may be any URI for data-transmission specifications, such as SMTP, FTP, etc.

SoapBindingStyle enumeration (System.Web.Services.Description)
ToString

## Description

Specifies the type of action that occurs in the Web Service at the level to which this enumeration is applied.

This enumeration allows the user to specify whether the message being transmitted is procedure- or document-oriented. Procedure-oriented messages contain parameters and return values, while document-oriented messages contain documents.

**ToString** 

[C#] public const SoapBindingStyle Default;

[C++] public: const SoapBindingStyle Default;

[VB] Public Const Default As SoapBindingStyle

[JScript] public var Default : SoapBindingStyle;

Description

1	This value specifies that the action should use the default value appropriate
2	to that level of the WSDL file.
3	ToString
4	
5	[C#] public const SoapBindingStyle Document;
6	[C++] public: const SoapBindingStyle Document;
7	[VB] Public Const Document As SoapBindingStyle
8	[JScript] public var Document : SoapBindingStyle;
9	
10	Description
11	Specifies that the message to be transmitted contains documents.
12	ToString
13	
14	[C#] public const SoapBindingStyle Rpc;
15	[C++] public: const SoapBindingStyle Rpc;
16	[VB] Public Const Rpc As SoapBindingStyle
17	[JScript] public var Rpc : SoapBindingStyle;
18	
19	Description
20	Specifies that the message to be transmitted contains a call to a procedure.
21	RPC is an acronym for "remote procedure call."
22	SoapBindingUse enumeration (System.Web.Services.Description)
23	ToString
24	
25	
	••

1	
2	
3	Description
4	Specifies whether the message parts are encoded using given rules, or
5	define a concrete schema for the message.
6	ToString
7	
8	[C#] public const SoapBindingUse Default;
9	[C++] public: const SoapBindingUse Default;
10	[VB] Public Const Default As SoapBindingUse
11	[JScript] public var Default : SoapBindingUse;
12	
13	Description
14	Specifies an empty string ("") value for the corresponding XML attribute.
15	ToString
16	
17	[C#] public const SoapBindingUse Encoded;
18	[C++] public: const SoapBindingUse Encoded;
19	[VB] Public Const Encoded As SoapBindingUse
20	[JScript] public var Encoded : SoapBindingUse;
21	
22	Description
23	Specifies that the message parts are encoded using given encoding rules.
24	ToString
25	

```
[C#] public const SoapBindingUse Literal;
    [C++] public: const SoapBindingUse Literal;
3
    [VB] Public Const Literal As SoapBindingUse
    [JScript] public var Literal : SoapBindingUse;
5
6
    Description
           Specifies that the message parts represent a concrete schema.
           SoapBodyBinding class (System.Web.Services.Description)
           ToString
11
12
    Description
13
           Represents an extensibility element added to an
14
    System.Web.Services.Description.InputBinding or an
15
    System. Web. Services. Description. Output Binding. It specifies how messages,
16
    either abstract type definitions or concrete schema definitions, appear within the
17
    SOAP body element of the transmission. This class cannot be inherited.
18
           SoapBodyBinding
19
           Example Syntax:
20
           ToString
21
22
    [C#] public SoapBodyBinding();
23
    [C++] public: SoapBodyBinding();
24
```

1	[VB] Public Sub New()
2	[JScript] public function SoapBodyBinding();
3	Encoding
4	ToString
5	
6	[C#] public string Encoding {get; set;}
7	[C++] public:property String* get_Encoding();public:property void
8	set_Encoding(String*);
9	[VB] Public Property Encoding As String
10	[JScript] public function get Encoding(): String; public function set
11	Encoding(String);
12	
13	Description
14	Supplies a list of space-delimited URIs representing the encoding style (or
15	styles) to be used to encode the messages within the SOAP body.
16	The value of this property should be set only if the value of the
17	System.Web.Services.Description.SoapBodyBinding.Use property is Encoded.
18	Handled
19	Namespace
20	ToString
21	
22	
23	Description
24	Get or sets the URI for encoding of content not specifically defined by the
25	System.Web.Services.Description.SoapBodyBinding.Encoding property.

Description

This property will return an empty string ("") if the property value has not been set. The value should only be set if the value of the 2 System.Web.Services.Description.SoapBodyBinding.Use property is Encoded . 3 Parent 4 Parts 5 **ToString** 6 7 8 Description 9 Indicates which parts of the transmitted message appear somewhere within 10 the SOAP body portion of that transmission. 11 Some parts of the transmitted message can appear in portions other than the 12 SOAP body, such as when SOAP is used with a 13 System.Web.Services.Description.MimeMultipartRelatedBinding. 14 **PartsString** 15 **ToString** 16 17 [C#] public string PartsString {get; set;} 18 [C++] public: property String\* get PartsString(); public: property void 19 set PartsString(String\*); 20 [VB] Public Property PartsString As String 21 [JScript] public function get PartsString(): String; public function set 22 PartsString(String); 23 24

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Indicates which parts of the transmitted message appear somewhere within the SOAP body portion of that transmission.

Some parts of the transmitted message can appear in portions other than the SOAP body, such as when SOAP is used with a

 $System. Web. Services. Description. Mime Multipart Related Binding \ . \\$ 

Required

Use

**ToString** 

Description

Indicates whether the message parts are encoded based on specified encoding rules, or define the concrete schema of the message.

The System.Web.Services.Description.SoapBodyBinding.Namespace and System.Web.Services.Description.SoapBodyBinding.Encoding properties depend on the value of this property. Their values should be set only if the value of this property is Encoded .

SoapExtensionImporter class (System.Web.Services.Description)

**ToString** 

Description

SoapExtensionImporter

Example Syntax:

1	ToString
2	
3	[C#] protected SoapExtensionImporter();
4	[C++] protected: SoapExtensionImporter();
5	[VB] Protected Sub New()
6	[JScript] protected function SoapExtensionImporter();
7	ImportContext
8	ToString
9	
10	[C#] public SoapProtocolImporter ImportContext {get; set;}
11	[C++] public:property SoapProtocolImporter* get_ImportContext();public:
12	property void set_ImportContext(SoapProtocolImporter*);
13	[VB] Public Property ImportContext As SoapProtocolImporter
14	[JScript] public function get ImportContext() : SoapProtocolImporter;public
15	function set ImportContext(SoapProtocolImporter);
16	
17	Description
18	
19	ImportMethod
20	
21	[C#] public abstract void ImportMethod(CodeAttributeDeclarationCollection
22	metadata);
23	[C++] public: virtual void ImportMethod(CodeAttributeDeclarationCollection*
24	metadata) = 0;
25	[VB] MustOverride Public Sub ImportMethod(ByVal metadata As

1	CodeAttributeDeclarationCollection)
2	[JScript] public abstract function ImportMethod(metadata:
3	CodeAttributeDeclarationCollection);
4	
5	Description
6	
7	SoapExtensionReflector class (System.Web.Services.Description)
8	ToString
9	
10	
11	Description
12	
13	SoapExtensionReflector
14	Example Syntax:
15	ToString
16	
17	[C#] protected SoapExtensionReflector();
18	[C++] protected: SoapExtensionReflector();
19	[VB] Protected Sub New()
20	[JScript] protected function SoapExtensionReflector();
21	ReflectionContext
22	ToString
23	
24	[C#] public ProtocolReflector ReflectionContext {get; set;}
25	[C++] public:property ProtocolReflector* get_ReflectionContext();public:

	1	property void set_ReflectionContext(ProtocolReflector*);
	2	[VB] Public Property ReflectionContext As ProtocolReflector
	3	[JScript] public function get ReflectionContext(): ProtocolReflector; public
	4	function set ReflectionContext(ProtocolReflector);
	5	
	6	Description
	7	
	8	ReflectMethod
	9	
	10	[C#] public abstract void ReflectMethod();
	11	[C++] public: virtual void ReflectMethod() = 0;
TU	12	[VB] MustOverride Public Sub ReflectMethod()
	13	[JScript] public abstract function ReflectMethod();
15 15 15 15 15 15 15 15 15 15 15 15 15 1	14	
	15	Description
	16	
i.d	17	SoapFaultBinding class (System.Web.Services.Description)
	18	ToString
	19	
	20	
	21	Description
	22	Represents an extensibility element added to a
	23	System.Web.Services.Description.FaultBinding within a Web Service. It

specifies the contents of any SOAP fault message returned. This class cannot be

inherited.

1	SoapFaultBinding
2	Example Syntax:
3	ToString
4	
5	[C#] public SoapFaultBinding();
6	[C++] public: SoapFaultBinding();
7	[VB] Public Sub New()
8	[JScript] public function SoapFaultBinding();
9	Encoding
10	ToString
11	
12	[C#] public string Encoding {get; set;}
13	[C++] public:property String* get_Encoding();public:property void
14	set_Encoding(String*);
15	[VB] Public Property Encoding As String
16	[JScript] public function get Encoding(): String; public function set
17	Encoding(String);
18	
19	11
20	Gets or sets a URI representing the encoding style used to encode the
21	SOAP fault message.
22	
23	System. Web. Services. Description. Soap Fault Binding. Use property is Encoded.
24	Handled
25	Namespace

15

16

17

18

19

20

21

22

23

24

25

**ToString** 3 Description Get or sets the URI for encoding of content not specifically defined by the 5 System.Web.Services.Description.SoapFaultBinding.Encoding property. This property will return an empty string ("") if the property value has not 7 been set. The value should only be set if the value of the System. Web. Services. Description. Soap Fault Binding. Use property is Encoded. Parent 10 Required 11 Use 12 **ToString** 13

Description

Specifies whether the fault message is encoded using encoding rules specified by the **System.Web.Services.Description.SoapFaultBinding.Encoding** property, or is encapsulated within a concrete schema.

SoapHeaderBinding class (System.Web.Services.Description)

**ToString** 

Description

```
Represents an extensibility element added to an
1
    System.Web.Services.Description.InputBinding or an
2
    System. Web. Services. Description. Output Binding within a Web Service. This
3
    class cannot be inherited.
4
           SoapHeaderBinding
5
           Example Syntax:
6
           ToString
7
8
    [C#] public SoapHeaderBinding();
9
    [C++] public: SoapHeaderBinding();
10
    [VB] Public Sub New()
11
    [JScript] public function SoapHeaderBinding();
12
           Encoding
13
           ToString
14
15
    [C#] public string Encoding {get; set;}
16
    [C++] public: __property String* get_Encoding();public: __property void
17
    set Encoding(String*);
18
    [VB] Public Property Encoding As String
19
    [JScript] public function get Encoding(): String; public function set
20
    Encoding(String);
21
22
    Description
23
           Gets or sets a URI representing the encoding style used to encode the
24
    SOAP header.
```

	1	The value of this property should be set only if the value of the
	2	System.Web.Services.Description.SoapHeaderBinding.Use property is
	3	Encoded .
	4	Handled
	5	MapToProperty
	6	ToString
	7	
	8	
	9	Description
	10	
and July Long Turn And Holl Link And	11	Message
	12	ToString
	13	
Fig.	14	[C#] public XmlQualifiedName Message {get; set;}
	15	[C++] public:property XmlQualifiedName* get_Message();public:property
	16	<pre>void set_Message(XmlQualifiedName*);</pre>
•	17	[VB] Public Property Message As XmlQualifiedName
	18	[JScript] public function get Message(): XmlQualifiedName; public function set
	19	Message(XmlQualifiedName);
	20	
	21	Description
	22	Gets or sets a value specifying the name of the
	23	System.Web.Services.Description.Message within the Web Service to which the
	24	SoapHeaderBinding applies.
	25	The default value of this property is an empty string ("").

1	Namespace
2	ToString
3	
4	[C#] public string Namespace {get; set;}
5	[C++] public:property String* get_Namespace();public:property void
6	set_Namespace(String*);
7	[VB] Public Property Namespace As String
8	[JScript] public function get Namespace() : String; public function set
9	Namespace(String);
10	
11	Description
12	Get or sets the URI for encoding of content not specifically defined by the
13	System.Web.Services.Description.SoapHeaderBinding.Encoding property.
14	This property will return an empty string ("") if the property value has not
15	been set. The value should only be set if the value of the
16	System.Web.Services.Description.SoapHeaderBinding.Use property is
17	Encoded .
18	Parent
19	Part
20	ToString
21	
22	
23	Description
24	
25	

Gets or sets a value indicating which 1 System. Web. Services. Description. Message Part within the Web Service the 2 SoapHeaderBinding applies to. Required Use **ToString** Description Specifies whether the header is encoded using encoding rules specified by 10 the System. Web. Services. Description. Soap Header Binding. Encoding property, 11 or is encapsulated within a concrete schema. 12 SoapHeaderFaultBinding class (System.Web.Services.Description) 13 **ToString** 14 15 16 Description 17 Represents an extensibility element added to an 18 System.Web.Services.Description.InputBinding or an 19 System. Web. Services. Description. Output Binding within the Web Service. It 20 specifies the SOAP header types used to transmit error information wihtin the 21 SOAP header. This class cannot be inherited. 22 SoapHeaderFaultBinding 23 Example Syntax: 24 **ToString** 25

```
[C#] public SoapHeaderFaultBinding();
    [C++] public: SoapHeaderFaultBinding();
    [VB] Public Sub New()
    [JScript] public function SoapHeaderFaultBinding();
           Encoding
           ToString
 8
    [C#] public string Encoding {get; set;}
 9
    [C++] public: __property String* get_Encoding();public: _ property void
10
    set_Encoding(String*);
11
    [VB] Public Property Encoding As String
12
    [JScript] public function get Encoding(): String; public function set
13
    Encoding(String);
14
15
    Description
16
           Gets or sets a URI representing the encoding style used to encode the error
17
    message for the SOAP header.
18
           The value of this property should be set only if the value of the
19
    System.Web.Services.Description.SoapHeaderFaultBinding.Use property is
20
    Encoded.
21
           Handled
22
           Message
23
           ToString
24
25
```

7

8

9

17

18

19

20

21

22

23

Description

Gets or sets a value specifying the name of the

System. Web. Services. Description. Message within the Web Service to which the SoapHeaderFaultBinding applies.

The default value of this property is an empty string ("").

Namespace

**ToString** 

[C#] public string Namespace {get; set;}

[C++] public: \_\_property String\* get\_Namespace();public: \_\_property void set Namespace(String\*);

[VB] Public Property Namespace As String

[JScript] public function get Namespace(): String; public function set Namespace(String);

Description

Get or sets the URI for encoding of content not specifically defined by the System.Web.Services.Description.SoapHeaderFaultBinding.Encoding property.

This property will return an empty string ("") if the property value has not been set. The value should only be set if the value of the

System.Web.Services.Description.SoapHeaderBinding.Use property is Encoded.

1	Parent
2	Part
3	ToString
4	
5	
6	Description
7	Gets or sets a value indicating which
8	System.Web.Services.Description.MessagePart within the Web Service the
9	SoapHeaderFaultBinding applies to.
10	Required
11	Use
12	ToString
13	
14	
15	Description
16	Specifies whether the header is encoded using encoding rules specified by
17	the System.Web.Services.Description.SoapHeaderBinding.Encoding property
18	or is encapsulated within a concrete schema.
19	SoapOperationBinding class (System.Web.Services.Description)
20	ToString
21	
22	
23	Description
24	Represents an extension added to an
25	System. Web. Services, Description, Operation Binding within a Web Service

	1	Specifies that the message transmission will be in SOAP format. This class canno
	2	be inherited.
	3	SoapOperationBinding
	4	Example Syntax:
	5	ToString
	6	
	7	[C#] public SoapOperationBinding();
	8	[C++] public: SoapOperationBinding();
	9	[VB] Public Sub New()
Sud Rus	10	[JScript] public function SoapOperationBinding();
ուսու մարդ անութ անութե հիոգի <u>հրութ</u>	11	Handled
Turbig, printer.	12	Parent
	13	Required
	14	SoapAction
	15	ToString
	16	
	17	
	18	Description
	19	Specifies the URI for the SOAP header.
	20	This property is required for HTTP protocol binding of SOAP.
	21	Style
	22	ToString
	23	
	24	[C#] public SoapBindingStyle Style {get; set;}
	25	[C++] public: property SoapBindingStyle get_Style():public: property void

```
set Style(SoapBindingStyle);
    [VB] Public Property Style As SoapBindingStyle
2
    [JScript] public function get Style(): SoapBindingStyle; public function set
    Style(SoapBindingStyle);
    Description
           One of the System. Web. Services. Description. Soap Binding Style values.
    The default is Document.
           SoapProtocolImporter class (System.Web.Services.Description)
9
           ToString
10
11
12
    Description
13
14
           SoapProtocolImporter
15
           Example Syntax:
16
           ToString
17
18
    [C#] public SoapProtocolImporter();
    [C++] public: SoapProtocolImporter();
20
    [VB] Public Sub New()
21
    [JScript] public function SoapProtocolImporter();
22
           AbstractSchemas
23
           Binding
24
           ClassName
25
```

1	ClassNames
2	CodeNamespace
3	CodeTypeDeclaration
4	ConcreteSchemas
5	InputMessage
6	MethodName
7	Operation
8	OperationBinding
9	OutputMessage
10	Port
11	PortType
12	ProtocolName
13	ToString
14	
15	
16	Description
17	
18	Schemas
19	Service
20	ServiceDescriptions
21	SoapBinding
22	ToString
23	
24	

ıs Description lee@hayes pilc 509+324+9256

	1	
	1	
	2	SoapExporter
	3	. ToString
	4	
	5	[C#] public SoapCodeExporter SoapExporter {get;}
	6	[C++] public:property SoapCodeExporter* get_SoapExporter();
	7	[VB] Public ReadOnly Property SoapExporter As SoapCodeExporter
	8	[JScript] public function get SoapExporter() : SoapCodeExporter;
	9	
	10	Description
	11	
٠.	12	SoapImporter
	13	ToString
	14	
	Ì	[C#] public SoapSchemaImporter SoapImporter {get;}
	15	
	16	[C++] public:property SoapSchemaImporter* get_SoapImporter();
	17	[VB] Public ReadOnly Property SoapImporter As SoapSchemaImporter
	18	[JScript] public function get SoapImporter(): SoapSchemaImporter;
	19	
	20	Description
	21	
	22	Style
	23	Warnings
	24	XmlExporter
	25	ToString
	ı	I and the state of

MS1-863US.APP

	,	
	2	
		Description
	3	Description
	4	37 1T .
	5	XmlImporter
	6	ToString
	7	
	8	[C#] public XmlSchemaImporter XmlImporter {get;}
	9	[C++] public:property XmlSchemaImporter* get_XmlImporter();
ing ing inf	10	[VB] Public ReadOnly Property XmlImporter As XmlSchemaImporter
	11	[JScript] public function get XmlImporter(): XmlSchemaImporter;
7 7000	12	
	13	Description
	14	
	15	BeginClass
	16	
	17	[C#] protected override CodeTypeDeclaration BeginClass();
	18	[C++] protected: CodeTypeDeclaration* BeginClass();
	19	[VB] Overrides Protected Function BeginClass() As CodeTypeDeclaration
	∦	[JScript] protected override function BeginClass(): CodeTypeDeclaration;
	20	[15cmpt] protected override function begine ass(). Code i ypebeciaration,
	21	
	22	Description
	23	
	24	BeginNamespace
	25	

1	
2	[C#] protected override void BeginNamespace();
3	[C++] protected: void BeginNamespace();
4	[VB] Overrides Protected Sub BeginNamespace()
5	[JScript] protected override function BeginNamespace()
6	
7	Description
8	
9	EndClass
10	
11	[C#] protected override void EndClass();
12	[C++] protected: void EndClass();
13	[VB] Overrides Protected Sub EndClass()
14	[JScript] protected override function EndClass();
15	
16	Description
17	
18	EndNamespace
19	
20	[C#] protected override void EndNamespace();
21	[C++] protected: void EndNamespace();
22	[VB] Overrides Protected Sub EndNamespace()
23	[JScript] protected override function EndNamespace();
24	
25	Description

2	GenerateMethod
3	
4	[C#] protected override CodeMemberMethod GenerateMethod();
5	[C++] protected: CodeMemberMethod* GenerateMethod();
6	[VB] Overrides Protected Function GenerateMethod() As CodeMemberMethod
7	[JScript] protected override function GenerateMethod(): CodeMemberMethod;
8	
9	Description
10	
11	IsBindingSupported
12	
13	[C#] protected override bool IsBindingSupported();
14	[C++] protected: bool IsBindingSupported();
15	[VB] Overrides Protected Function IsBindingSupported() As Boolean
16	[JScript] protected override function IsBindingSupported(): Boolean;
17	
18	Description
19	
20	IsOperationFlowSupported
21	
22	[C#] protected override bool IsOperationFlowSupported(OperationFlow flow);
23	[C++] protected: bool IsOperationFlowSupported(OperationFlow flow);

[VB] Overrides Protected Function IsOperationFlowSupported(ByVal flow As

OperationFlow) As Boolean

	1	[JScript] protected override function IsOperationFlowSupported(flow:			
	2	OperationFlow): Boolean;			
	3				
	4	Description			
	5				
	6	SoapTransportImporter class (System.Web.Services.Description)			
	7	UnsupportedOperationWarning			
	8				
	9				
	10	Description			
	11				
	12	SoapTransportImporter			
	13	Example Syntax:			
	14	UnsupportedOperationWarning			
H- H.F H.F H- M L.F	15				
	16	[C#] protected SoapTransportImporter();			
ala ala	17	[C++] protected: SoapTransportImporter();			
	18	[VB] Protected Sub New()			
	19	[JScript] protected function SoapTransportImporter();			
	20	ImportContext			
	21	UnsupportedOperationWarning			
	22				
	23	[C#] public SoapProtocolImporter ImportContext {get; set;}			
	24	[C++] public:property SoapProtocolImporter* get_ImportContext();public:			
	25	property void set_ImportContext(SoapProtocolImporter*);			

	1	[VB] Public Property ImportContext As SoapProtocolImporter
	2	[JScript] public function get ImportContext() : SoapProtocolImporter;public
	3	function set ImportContext(SoapProtocolImporter);
	4	
	5	Description
	6	
	7	ImportClass
	8	
	9	[C#] public abstract void ImportClass();
	10	[C++] public: virtual void ImportClass() = 0;
	11	[VB] MustOverride Public Sub ImportClass()
	12	[JScript] public abstract function ImportClass();
	13	
	14	Description
	15	
	16	IsSupportedTransport
i dia	17	
	18	[C#] public abstract bool IsSupportedTransport(string transport);
	19	[C++] public: virtual bool IsSupportedTransport(String* transport) = 0;
	20	[VB] MustOverride Public Function IsSupportedTransport(ByVal transport As
	21	String) As Boolean
	22	[JScript] public abstract function IsSupportedTransport(transport : String) :
	23	Boolean;
	24	
	25	Description

2	
3	
4	
5	
6	
7	
8	
0	
,,	
10	1
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	

Types class (System.Web.Services.Description)
ToString

# Description

Derived from the **System.Web.Services.Description.DocumentableItem** class, this class describes data type definitions relevant to exchanged messages.

This class cannot be inherited.

This class is a container for datatype definitions for the Web Service. Note especially that this class does NOT represent a collection of **System.Type** objects, as its name might imply.

Types

Example Syntax:

ToString

Extensions

**ToString** 

25

# Description

Gets the collection of ServiceDescriptionFormatExtension elements incuded in the Web Service. This property is read-only.

In the default implementation of the

# System.Web.Services.Discovery

# Description

The **System.Web.Services.Discovery** namespace consists of the classes that allows Web Service consumers to locate the available Web Services on a Web server through a process called Web Services Discovery.

ContractReference class (System.Web.Services:Discovery)

# Description

Represents a reference in a discovery document to a Service Description.

Web Services discovery involves discovering the available Web Services given an URL. The URL usually points to a discovery document, which typically has a .disco file extension. Within a discovery document are references to information about the existance of Web Services. These references can refer to

3

4

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Service Descriptions, XML Schema Definition language (XSD) schemas or other discovery documents. This class represents a reference to a Service Description.

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;

Description

XML namespace for Service Description references in discovery documents.

Within a discovery document, a reference to a Service Description is contained within a **contractRef** XML element, which is a part of the XML namespace specified in the

 $System. Web. Services. Discovery. Contract Reference. Names pace \ {\tt constant}.$ 

Constructors:

ContractReference

Example Syntax:

[C#] public ContractReference();

[C++] public: ContractReference();

[VB] Public Sub New()

[JScript] public function ContractReference(); Initializes a new instance of the

System.Web.Services.Discovery.ContractReference class.

> 18 19

> 15

16

17

20 21

22

23

24 25 Description

1

2

3

5

7

9

Initializes a new instance of the

System. Web. Services. Discovery. Contract Reference class using default values.

Use this constructor to create and initialize a new instance of the System.Web.Services.Discovery.ContractReference class using default values.

ContractReference

Example Syntax:

[C#] public ContractReference(string href);

[C++] public: ContractReference(String\* href);

[VB] Public Sub New(ByVal href As String)

[JScript] public function ContractReference(href: String);

Description

Initializes a new instance of the

System.Web.Services.Discovery.ContractReference class using the supplied reference to a Service Description.

Use this constructor to create and initialize a new instance of the System. Web. Services. Discovery. Contract Reference class using the specified reference name. The URL for a Sevice Descritpion. Initializes the  ${\bf System. Web. Services. Discovery. Contract Reference. Ref \ property \ value.}$ 

ContractReference

Example Syntax:

1

5

18

15

16

17

20

19

22

21

23

2425

[C#] public ContractReference(string href, string docRef);

[C++] public: ContractReference(String\* href, String\* docRef);

[VB] Public Sub New(ByVal href As String, ByVal docRef As String)

[JScript] public function ContractReference(href: String, docRef: String);

Description

Initializes a new instance of the

System.Web.Services.Discovery.ContractReference class using the supplied reference to a Service Description and a Web Service implementing the Service Description.

Use this constructor to create and initialize a new instance of the System.Web.Services.Discovery.ContractReference class using the specified reference name and docRef. The URL for a Service Description. Initializes the System.Web.Services.Discovery.ContractReference.Ref property value. The URL for a Web Service implementing the Service Description at *href*. Initializes the System.Web.Services.Discovery.ContractReference.DocRef property value.

Properties:

ClientProtocol

Contract

Description

Gets a **System.Web.Services.Description.ServiceDescription** object representing the Service Description.

2

3

7

8

9

10

11

12

13

14

15

17

18

19

20

21

22

23

25

1	Defai	1tFil	lenam	ρ
	Jerau	ше	ichani	C

[C#] public override string DefaultFilename {get;}

[C++] public: \_\_property virtual String\* get\_DefaultFilename();

[VB] Overrides Public ReadOnly Property DefaultFilename As String

[JScript] public function get DefaultFilename(): String;

#### Description

Gets the name of the file to use by default when saving the referenced Service Description.

DocRef

[C#] public string DocRef {get; set;}

[C++] public: \_\_property String\* get\_DocRef();public: \_\_property void set DocRef(String\*);

[VB] Public Property DocRef As String

[JScript] public function get DocRef(): String; public function set DocRef(String);

# Description

Gets and sets the URL for a Web Service implementing the Service Description referenced in the

 ${\bf System. Web. Services. Discovery. Contract Reference. Ref \ property.}$ 

Within a discovery document, a reference to a Service Description is contained within a **contractRef** XML element. The **contractRef** XML element has two attributes described in the following table.

Ref 2 [C#] public string Ref {get; set;} 3 [C++] public: \_\_property String\* get\_Ref();public: \_\_property void set Ref(String\*); 5 [VB] Public Property Ref As String [JScript] public function get Ref(): String; public function set Ref(String); 7 8 Description 9 Gets or sets the URL to the referenced Service Description. 10 Within a discovery document, a reference to a Service Description is 11 contained within a contractRef XML element. The contractRef XML element has 12 two attributes described in the following table. 13 Url 14 15 [C#] public override string Url {get; set;} 16 [C++] public: \_\_property virtual String\* get\_Url();public: \_\_property virtual void 17 set Url(String\*); 18 [VB] Overrides Public Property Url As String 19 [JScript] public function get Url(): String; public function set Url(String); 20 21 Description 22 Gets or sets the URL for the referenced Service Description. 23 24 25

18

19

20

21

22

23

25

1

2

3

5

6

For the System. Web. Services. Discovery. Contract Reference class, the System.Web.Services.Discovery.ContractReference.Url property returns the value of the System. Web. Services. Discovery. Contract Reference. Ref property.

Methods:

ReadDocument

[C#] public override object ReadDocument(Stream stream);

[C++] public: Object\* ReadDocument(Stream\* stream);

[VB] Overrides Public Function ReadDocument(ByVal stream As Stream) As Object

[JScript] public override function ReadDocument(stream : Stream) : Object;

# Description

Reads the Service Description from the passed System.IO.Stream and returns the Service Description.

Return Value: A System. Web. Services. Description. Service Description containing the contents of the referenced Service Description. System.IO.Stream containing the Service Description.

Resolve

[C#] protected internal override void Resolve(string contentType, Stream stream);

[C++] protected public: void Resolve(String\* contentType, Stream\* stream);

[VB] Overrides Protected Friend Dim Sub Resolve(ByVal contentType As String,

ByVal stream As Stream)

[JScript] package override function Resolve(contentType : String, stream :

14 15

> 16 17

> > 18

19

20

22

21

23

24

25

Stream);

2

Description

Resolves whether the the referenced document is valid.

If the MIME type is text/xml and the contents of stream is a Service Description, then the contents of stream are added to the

 $System. Web. Services. Discovery. Discovery Client Protocol. References \ and \ an$ System. Web. Services. Discovery. Discovery Client Protocol. Documentsproperties of

 $System. Web. Services. Discovery. Discovery Reference. Client Protocol\ .\ The$ MIME content type of stream. The System.IO.Stream containing the referenced document.

WriteDocument

[C#] public override void WriteDocument(object document, Stream stream); [C++] public: void WriteDocument(Object\* document, Stream\* stream); [VB] Overrides Public Sub WriteDocument(ByVal document As Object, ByVal stream As Stream)

[JScript] public override function WriteDocument(document : Object, stream : Stream);

Description

Writes the passed-in Service Description into the passed-in System.IO.Stream . The System.Web.Services.Description.ServiceDescription

	1	to write into stream. The System.IO.Stream into which the serialized
	2	System.Web.Services.Description.ServiceDescription is written.
	3	ContractSearchPattern class (System.Web.Services.Discovery)
	4	WriteDocument
	5	
	6	
	7	Description
	8	Represents a contract search pattern. This class cannot be inherited.
	9	ContractSearchPattern
: 22	10	Example Syntax:
	11	WriteDocument
	12	
	13	[C#] public ContractSearchPattern();
	14	[C++] public: ContractSearchPattern();
The state of the s	15	[VB] Public Sub New()
	16	[JScript] public function ContractSearchPattern();
Franks	17	Pattern
	18	WriteDocument
	19	
	20	[C#] public override string Pattern {get;}
	21	[C++] public:property virtual String* get_Pattern();
	22	[VB] Overrides Public ReadOnly Property Pattern As String
	23	[JScript] public function get Pattern(): String;
	24	
	25	Description

20

21

22

23

24

Overrides the base implementation	to return	the literal	string,	"*.asmx".
GetDiscoveryReference				

[C#] public override DiscoveryReference GetDiscoveryReference(string filename);

[C++] public: DiscoveryReference\* GetDiscoveryReference(String\* filename);

[VB] Overrides Public Function GetDiscoveryReference(ByVal filename As

String) As DiscoveryReference

[JScript] public override function GetDiscoveryReference(filename : String) : DiscoveryReference;

#### Description

Overrides the base implementation to create a **System.Web.Services.Discovery.ContractReference** object with the specified filename.

Return Value: A System.Web.Services.Discovery.ContractReference object with the specified filename. A string that represents a filename.

DiscoveryClientDocumentCollection class

(System.Web.Services.Discovery)

**ToString** 

# Description

Represents a collection of documents discovered during Web Services discovery that have been downloaded to the client. This class cannot be inherited.

716

25

Keys

The 1 System.Web.Services.Discovery.DiscoveryClientProtocol.Documents property 2 of System. Web. Services. Discovery. Discovery Client Protocol is of type 3 System. Web. Services. Discovery. Discovery Client Document Collection.DiscoveryClientDocumentCollection 5 Example Syntax: 6 **ToString** 7 8 [C#] public DiscoveryClientDocumentCollection(); 9 [C++] public: DiscoveryClientDocumentCollection(); 10 [VB] Public Sub New() 11 [JScript] public function DiscoveryClientDocumentCollection(); 12 Count 13 Dictionary 14 InnerHashtable 15 Item **ToString** 17 18 19 Description 20 Gets or sets a client discovery document object from the 21 System.Web.Services.Discovery.DiscoveryClientDocumentCollection with the 22 specified URL. The URL of the discovery document to get or set from the 23 System.Web.Services.Discovery.DiscoveryClientDocumentCollection.

1	ToString
2	
3	[C#] public ICollection Keys {get;}
4	[C++] public:property ICollection* get_Keys();
5	[VB] Public ReadOnly Property Keys As ICollection
6	[JScript] public function get Keys(): ICollection;
7	
8	Description
9	Gets an System.Collections.ICollection object with all of the keys in the
10	System.Web.Services.Discovery.DiscoveryClientDocumentCollection .
] ] 11	Values
10 11 11 12 13 13 13 13 13 13 13 13 13 13 13 13 13	ToString
13 13	
14	[C#] public ICollection Values {get;}
15	[C++] public:property ICollection* get_Values();
16	[VB] Public ReadOnly Property Values As ICollection
17	[JScript] public function get Values(): ICollection;
18	
19	Description
20	Gets an System.Collections.ICollection object with all of the values in the
21	System.Web.Services.Discovery.DiscoveryClientDocumentCollection .
22	Add
23	
24	[C#] public void Add(string url, object value);
25	[C++] public: void Add(String* url, Object* value);

[VB] Public Sub Add(ByVal url As String, ByVal value As Object) [JScript] public function Add(url : String, value : Object); 2 3 Description Adds an object with the specified URL to the System.Web.Services.Discovery.DiscoveryClientDocumentCollection. The URL for the document to add to the System.Web.Services.Discovery.DiscoveryClientDocumentCollection . A discovered document to add to the System.Web.Services.Discovery.DiscoveryClientDocumentCollection. **Contains** [C#] public bool Contains(string url); [C++] public: bool Contains(String\* url); [VB] Public Function Contains(ByVal url As String) As Boolean [JScript] public function Contains(url : String) : Boolean; Description Determines if the 19 System.Web.Services.Discovery.DiscoveryClientDocumentCollection contains 20 an object with the specified URL. 21 Return Value: true if the 22 System.Web.Services.Discovery.DiscoveryClientDocumentCollection contains 23 an object with the specified URL; otherwise, false. The URL for the document to 24

19

20

21

22

23

24

25

2

3

5

6

locate within the

System. Web. Services. Discovery. Discovery Client Document Collection.

Remove

[C#] public void Remove(string url);

[C++] public: void Remove(String\* url);

[VB] Public Sub Remove(ByVal url As String)

[JScript] public function Remove(url : String);

#### Description

Removes an object with the specified URL from the

System.Web.Services.Discovery.DiscoveryClientDocumentCollection . The

URL for the discovered document to remove from the

System.Web.Services.Discovery.DiscoveryClientDocumentCollection.

DiscoveryClientProtocol class (System.Web.Services.Discovery)

**ToString** 

# Description

Provides support for programmatically invoking Web Services discovery.

Web Service discovery is the process of locating, or discovering, one or more related documents that describe available Web Services. It is through Web Services discovery that Web Service clients learn about the available Web Services at a given URL and how to use them. Web Services discovery works from the premise that you have already obtained the URL to a discovery

```
document, possibly through a directory service, such as http://uddi.microsoft.com,
       however, you do not have the details about the Web Services offered. Through
   2
       Web Services discovery, you can discover the details about the Web Services
   3
       listed in a System. Web. Services. Discovery. Discovery Document at a specific
       URL.
              DiscoveryClientProtocol
   6
              Example Syntax:
              ToString
   8
   9
       [C#] public DiscoveryClientProtocol();
111 12 12 13 13
       [C++] public: DiscoveryClientProtocol();
       [VB] Public Sub New()
       [JScript] public function DiscoveryClientProtocol();
14
       Description
              Initializes a new instance of the
       System.Web.Services.Discovery.DiscoveryClientProtocol class.
   17
              AdditionalInformation
   18
              ToString
   19
   20
       [C#] public IList AdditionalInformation {get;}
   21
       [C++] public: __property IList* get AdditionalInformation();
  22
       [VB] Public ReadOnly Property AdditionalInformation As IList
  23
       [JScript] public function get AdditionalInformation(): IList;
  24
```

System on pro

21

22

23 24

25

Description

Gets information in addition to references found in the discovery document.

The

System. Web. Services. Discovery. Discovery Client Protocol. Additional Information of the protocol of the p

on property primarily contains SOAP bindings, represented by the

**System.Web.Services.Discovery.SoapBinding** class, defined in the discovery document.

AllowAutoRedirect

ClientCertificates

ConnectionGroupName

Container

CookieContainer

Credentials

DesignMode

**Documents** 

**ToString** 

Description

Gets a collection of discovery documents.

The

System. Web. Services. Discovery. Discovery Client Protocol. Documents

collection is populated during invocations to the

System. Web. Services. Discovery. Discovery Client Protocol. Discover (System. Structure) and the protocol of the protocol o

1	ing),
2	System. We b. Services. Discovery. Discovery Client Protocol. Discover Any (System) and the protocol of the
3	m.String),
4	$System. Web. Services. Discovery. Discovery Client Protocol. Resolve All\ and$
5	System. We b. Services. Discovery. Discovery Client Protocol. Resolve One Level
6	methods. During invocations to the
7	System. Web. Services. Discovery. Discovery Client Protocol. Discover (System. Structure) and the protocol of the protocol o
8	ing) and
9	System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste
10	m.String) methods, if the supplied URL is a valid discovery document, that
] ] 11	document is added to the
10	System.Web.Services.Discovery.DiscoveryClientProtocol.Documents
13	collection. During invocations to the
14	System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll and
15	System. Web. Services. Discovery. Discovery Client Protocol. Resolve One Level
16	methods, valid discovery document references in the
17	System.Web.Services.Discovery.DiscoveryClientProtocol.References
18	collection are added to the
19	System.Web.Services.Discovery.DiscoveryClientProtocol.Documents
20	collection.
21	Errors
22	ToString
23	
24	[C#] public DiscoveryExceptionDictionary Errors {get;}
25	[C++] public:property DiscoveryExceptionDictionary* get_Errors();

20

21

22

23

24

25

[VB] Public ReadOnly Property Errors As DiscoveryExceptionDictionary [JScript] public function get Errors() : DiscoveryExceptionDictionary;

Description

1

2

3

5

6

7

8

9

Gets a collection of exceptions that occurred during invocation of method from this class.

The System.Web.Services.Discovery.DiscoveryClientProtocol.Errors collection is populated with exceptions that occurred during invocations to the System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.String),

 $System. Web. Services. Discovery. Discovery Client Protocol. Discover Any (System. String) \ ,$ 

System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll, and System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel methods. The System.Web.Services.Discovery.DiscoveryExceptionDictionary is cleared on invocation of these methods.

**Events** 

PreAuthenticate

**Proxy** 

References

**ToString** 

Description

A collection of references founds in resolved discovery documents.

20

21

22

23

24

25

1	The
2	System.Web.Services.Discovery.DiscoveryClientProtocol.References
3	collection is populated during invocations to the
4	System. We b. Services. Discovery. Discovery Client Protocol. Discover (System. Structure) and the protocol of the protocol
5	ing),
6	System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste
7	m.String),
8	System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll and
9	System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel
10	methods. During invocations to the
10   11   12   13   13   13   13   13   13	System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.Str
12	ing) and
13	System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste
14	m.String) methods, if the supplied URL is a valid discovery document, that
15	document is added to the
16	System.Web.Services.Discovery.DiscoveryClientProtocol.References
17	collection together with references found in the discovery document. References
18	added during invocations to the

System. Web. Services. Discovery. Discovery Client Protocol. Discover (System. Structure) and the protocol of the protocol oing) and

System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste m.String) methods are not necessarily valid discovery documents. During invocations to

System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll and System. Web. Services. Discovery. Discovery Client Protocol. Resolve One Level

MS1-863US.APP

23

references within the System.Web.Services.Discovery.DiscoveryClientProtocol.References collection are verified as being valid discovery documents. If they are valid and contain references, they are also added to the System. Web. Services. Discovery. Discovery Client Protocol. Referencescollection. RequestEncoding Site **Timeout** Url UserAgent Discover [C#] public DiscoveryDocument Discover(string url); [C++] public: DiscoveryDocument\* Discover(String\* url); [VB] Public Function Discover(ByVal url As String) As DiscoveryDocument [JScript] public function Discover(url : String) : DiscoveryDocument; Description Discovers the supplied URL to determine if it is a discovery document. 20 21

Return Value: A System. Web. Services. Discovery. Discovery Document containing the results of Web Services discovery at the supplied URL.

The

System.Web.Services.Discovery.DiscoveryClientProtocol.Discover(System.Str ing) method expects that the supplied URL is a discovery document. If the URL

refers to a Service Description or an XML schema an exception is thrown. To 1 discover an XML schema or a Service Description invoke the 2

18

20

21

19

22 23

24

25

System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste m.String) method. The URL where Web Services discovery begins. DiscoverAny

[C#] public DiscoveryDocument DiscoverAny(string url); [C++] public: DiscoveryDocument\* DiscoverAny(String\* url); [VB] Public Function DiscoverAny(ByVal url As String) As DiscoveryDocument [JScript] public function DiscoverAny(url : String) : DiscoveryDocument;

Description

Service Description or an XML Schema Definition (XSD) schema. Return Value: A System. Web. Services. Discovery. Discovery Document containing the results of Web Services discovery at the supplied URL. If the url parameter refers to a Service Description or an XSD Schema, a System.Web.Services.Discovery.DiscoveryDocument is created in memory for it.

Discovers the supplied URL to determine if it is a discovery document,

The

System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoverAny(Syste m.String) method discovers whether the supplied URL is a discovery document, Service Description or XSD schema. If it is known that the URL only refers to a discovery document, the

1	System. Web. Services. Discovery. Discovery Client Protocol. Discover (System. Structure) and the protocol of the protocol o
2	ing) method might be invoked. The URL where Web Services discovery begins.
3	Download
4	
5	[C#] public Stream Download(ref string url);
6	[C++] public: Stream* Download(String** url);
7	[VB] Public Function Download(ByRef url As String) As Stream
8	[JScript] public function Download(url : String) : Stream; Downloads the
9	discovery document at the supplied URL into a System.IO.Stream object.
10	
11	Description
12	Downloads the discovery document at the supplied URL into a
13	System.IO.Stream object.
14	Return Value: A System.IO.Stream containing the document at the supplied
15	URL. The URL of the discovery document to download.
16	Download
17	
18	[C#] public Stream Download(ref string url, ref string contentType);
19	[C++] public: Stream* Download(String** url, String** contentType);
20	[VB] Public Function Download(ByRef url As String, ByRef contentType As
21	String) As Stream
22	[JScript] public function Download(url : String, contentType : String) : Stream;
23	
24	Description
25	

20

21

22

23

24

25

Downloads the discovery document at the supplied URL into a **System.IO.Stream** object, setting the *contentType* parameter to the MIME encoding of the discovery document.

Return Value: A System.IO.Stream containing the document at the supplied URL. The URL of the discovery document to download. The MIME encoding of the downloaded discovery document.

ReadAll

[C#] public DiscoveryClientResultCollection ReadAll(string topLevelFilename);

[C++] public: DiscoveryClientResultCollection\* ReadAll(String\* topLevelFilename);

[VB] Public Function ReadAll(ByVal topLevelFilename As String) As

DiscoveryClientResultCollection

[JScript] public function ReadAll(topLevelFilename : String) :

DiscoveryClientResultCollection;

Description

Reads in a file containing a map of saved discovery documents populating the System.Web.Services.Discovery.DiscoveryClientProtocol.Documents and System.Web.Services.Discovery.DiscoveryClientProtocol.References properties, with discovery documents, XML Schema Definition (XSD) schemas, and Service Descriptions referenced in the file.

Return Value: A

System.Web.Services.Discovery.DiscoveryClientResultCollection containing the results found in the file with the map of saved discovery documents. The file

21

22

23

24

25

format is a

System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoveryClientRes ultsFile class serialized into XML; however, one would typically create the file using only the

System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.String,System.String) method or Disco.exe.

A file containing a map of saved discovery documents can be created by the System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.String,System.String) method or Disco.exe. Name of file to read in, containing the map of saved discovery documents.

ResolveAll

[C#] public void ResolveAll();

[C++] public: void ResolveAll();

[VB] Public Sub ResolveAll()

[JScript] public function ResolveAll();

Description

Resolves all references to discovery documents, XML Schema Definition (XSD) schemas, and Service Descriptions in the

**System.Web.Services.Discovery.DiscoveryClientProtocol.References** property, as well as references found in referring discovery documents.

System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll resolves all valid references it finds and places them in the

System.Web.Services.Discovery.DiscoveryClientProtocol.Documents property.

Both System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll and System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel download and verify XSD schemas and Service Descriptions in the System.Web.Services.Discovery.DiscoveryClientProtocol.References property. However, the two methods differ in how they handle discovery documents.

ResolveOneLevel

[C#] public void ResolveOneLevel();

[C++] public: void ResolveOneLevel();

[VB] Public Sub ResolveOneLevel()

[JScript] public function ResolveOneLevel();

# Description

Resolves all references to discovery documents, XML Schema Definition (XSD) schemas and Service Descriptions in

System.Web.Services.Discovery.DiscoveryClientProtocol.References, as well as references found in those discovery documents.

System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneL evel resolves all valid references it finds and places them in the System.Web.Services.Discovery.DiscoveryClientProtocol.Documents property. Both System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll and System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel download and verify XSD schemas and Service Descriptions in the System.Web.Services.Discovery.DiscoveryClientProtocol.References property. However, the two methods differ in how they handle discovery documents.

,::::::
·Q
E
### ###
i gar
عادي
Ē
e raga e raga

V	(/ <sub>*</sub>	it	_	A	1	
v	VΙ	-11	•	$\boldsymbol{\vdash}$	ŀ	

3

5

6

7

9

10

11

12

13

14

15

17

18

20

22

23

1

[C#] public DiscoveryClientResultCollection WriteAll(string directory, string topLevelFilename);

[C++] public: DiscoveryClientResultCollection\* WriteAll(String\* directory,

String\* topLevelFilename);

[VB] Public Function WriteAll(ByVal directory As String, ByVal topLevelFilename As String) As DiscoveryClientResultCollection

[JScript] public function WriteAll(directory: String, topLevelFilename: String):

DiscoveryClientResultCollection;

# Description

Writes all discovery documents, XML Schema Definition (XSD) schemas, and Service Descriptions in the

**System.Web.Services.Discovery.DiscoveryClientProtocol.Documents** property to the supplied directory and creates a file in that directory.

Return Value: A

System.Web.Services.Discovery.DiscoveryClientResultCollection containing the results of all files saved.

The file created with the name of the *topLevelFilename* parameter in the directory specified by the directory parameter contains a map of saved discovery documents, XML Schema Definition (XSD) schemas and Service Descriptions. This file can be read in using the

System.Web.Services.Discovery.DiscoveryClientProtocol.ReadAll(System.String) method to populate the

1	System.Web.Services.Discovery.DiscoveryClientProtocol.References and					
2	System.Web.Services.Discovery.DiscoveryClientProtocol.Documents					
3	properties. The format of the file is XML containing an serialized version of the					
System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoveryC						
5	ultsFile class. The directory in which to save all documents currently in the					
6	System.Web.Services.Discovery.DiscoveryClientProtocol.Documents property.					
7	The name of the file to create or overwrite containing a map of all documents					
8	saved.					
9	DiscoveryClientReferenceCollection class					
I 10	(System.Web.Services.Discovery)					
10	WriteAll					
12						
13 H						
14 l	Description					
14   14   15	Represents a collection of					
16	System.Web.Services.Discovery.DiscoveryReference objects. This class cannot					
17	be inherited.					
18	The					
19	System.Web.Services.Discovery.DiscoveryClientProtocol.References property					
20	of System.Web.Services.Discovery.DiscoveryClientProtocol is of type					
21	System.Web.Services.Discovery.DiscoveryClientReferenceCollection .					
22	DiscoveryClientReferenceCollection					
23	Example Syntax:					
24	WriteAll					

	11
1	
2	[C#] public DiscoveryClientReferenceCollection();
3	[C++] public: DiscoveryClientReferenceCollection();
4	[VB] Public Sub New()
5	[JScript] public function DiscoveryClientReferenceCollection();
6	Count
7	Dictionary
8	InnerHashtable
9	Item
] 10	WriteAll
== 	
## 13	Description
1.	Gets or sets a System. Web. Services. Discovery. Discovery Reference
1:	object from the
1	System.Web.Services.Discovery.DiscoveryClientReferenceCollection with the
1 1	specified URL. The URL for the DiscoveryReference to get or set from the
1	System.Web.Services.Discovery.DiscoveryClientReferenceCollection.
1	9 Keys
2	WriteAll
2	
2	[C#] public ICollection Keys {get;}
2	[C++] public:property ICollection* get_Keys();
2	[VB] Public ReadOnly Property Keys As ICollection
2	[JScript] public function get Keys(): ICollection;

	1
ľ	1
T.	]
7	
	J
	1
	1
## ##	
	1
dist	

Description Gets an System.Collections.ICollection object with all of the keys in the 3  $System. Web. Services. Discovery. Discovery Client Reference Collection \ .$ Values 5 WriteAll 6 7 [C#] public ICollection Values {get;} [C++] public: property ICollection\* get\_Values(); [VB] Public ReadOnly Property Values As ICollection 0 [JScript] public function get Values(): ICollection; 2 Description 13 Gets an System. Collections. I Collection object with all of the values in the 14 System.Web.Services.Discovery.DiscoveryClientReferenceCollection . 15 Add 16 17 [C#] public void Add(DiscoveryReference value); 18 [C++] public: void Add(DiscoveryReference\* value); 19 [VB] Public Sub Add(ByVal value As DiscoveryReference) 20 [JScript] public function Add(value : DiscoveryReference); Adds a 21 System. Web. Services. Discovery. Discovery Reference to the 22  $System. Web. Services. Discovery. Discovery Client Reference Collection \ .$ 23 24 Description 25

	2	
	3	
	J	
	4	
	5	
	6	
	7	
	8	
	9	
100	10	
	11	
	12	
	13	
	14	
	15	
	16	
sdl	17	
	18	

20

21

22

23

24

1

Adds a System.Web.Services.Discovery.DiscoveryReference to the System.Web.Services.Discovery.DiscoveryClientReferenceCollection . The System.Web.Services.Discovery.DiscoveryReference to add to the System.Web.Services.Discovery.DiscoveryClientReferenceCollection.

Add

[C#] public void Add(string url, DiscoveryReference value);

[C++] public: void Add(String\* url, DiscoveryReference\* value);

[VB] Public Sub Add(ByVal url As String, ByVal value As DiscoveryReference)

[JScript] public function Add(url : String, value : DiscoveryReference);

Description

Adds a **System.Web.Services.Discovery.DiscoveryReference** with the specified URL and value to the

 $System. Web. Services. Discovery. Discovery Client Reference Collection \ . \ The$ 

URL for the reference to add to the

 $System. Web. Services. Discovery. Discovery Client Reference Collection. \ The$ 

DiscoveryReference to add to the

System. Web. Services. Discovery. Discovery Client Reference Collection.

Contains

[C#] public bool Contains(string url);

[C++] public: bool Contains(String\* url);

[VB] Public Function Contains(ByVal url As String) As Boolean

[JScript] public function Contains(url : String) : Boolean;

19

20

21

22

23

24

25

3

Description

Determines if the

System.Web.Services.Discovery.DiscoveryClientReferenceCollection contains a System.Web.Services.Discovery.DiscoveryReference with the specified URL.

Return Value: true if the

System.Web.Services.Discovery.DiscoveryClientReferenceCollection contains a System.Web.Services.Discovery.DiscoveryReference with the specified URL; otherwise, false. The URL for the

System.Web.Services.Discovery.DiscoveryReference to locate within the System.Web.Services.Discovery.DiscoveryClientReferenceCollection.

Remove

[C#] public void Remove(string url);

[C++] public: void Remove(String\* url);

[VB] Public Sub Remove(ByVal url As String)

[JScript] public function Remove(url : String);

Description

Removes a **System.Web.Services.Discovery.DiscoveryReference** with the specified URL from the

**System.Web.Services.Discovery.DiscoveryClientReferenceCollection** . A string that represents the URL for the object to remove from the

System. Web. Services. Discovery. Discovery Client Reference Collection.

DiscoveryClientResult class (System.Web.Services.Discovery)

1	ToString
2	
3	
4	Description
5	Represents the details of a discovery reference without the actual contents
6	of the referenced document. This class cannot be inherited.
7	Using the
8	System. Web. Services. Discovery. Discovery Client Protocol. Write All (System. Structure) and the protocol of the protocol
9	ing, System. String) method of
10	System.Web.Services.Discovery.DiscoveryClientProtocol, all valid referenced
11	documents and a file containing a map of all save documents can be written to
12	disk. The file containing a map of all saved documents contains the details of each
13	document as summarized by the properties of
14	System.Web.Services.Discovery.DiscoveryClientResult.
15	DiscoveryClientResult
16	Example Syntax:
17	ToString
18	
19	[C#] public DiscoveryClientResult();
20	[C++] public: DiscoveryClientResult();
21	[VB] Public Sub New()
22	[JScript] public function DiscoveryClientResult(); Initializes a new instance of the
23	System. Web. Services. Discovery. Discovery Client Result class.

25 Description

	1	Initializes a new instance of the
	2	System.Web.Services.Discovery.DiscoveryClientResult class.
	3	DiscoveryClientResult
	4	Example Syntax:
	5	ToString
	6	
	7	[C#] public DiscoveryClientResult(Type referenceType, string url, string
	8	filename);
	9	[C++] public: DiscoveryClientResult(Type* referenceType, String* url, String*
and the	10	filename);
July July	11	[VB] Public Sub New(ByVal referenceType As Type, ByVal url As String, ByVal
İ	12	filename As String)
	13	[JScript] public function DiscoveryClientResult(referenceType: Type, url: String
	14	filename : String);
	15	
	16	Description
	17	Initializes a new instance of the
	18	System.Web.Services.Discovery.DiscoveryClientResult class and sets the
	19	System.Web.Services.Discovery.DiscoveryClientResult.ReferenceTypeName
:	20	property to referenceType, the
	21	System.Web.Services.Discovery.DiscoveryClientResult.Url property to url and
:	22	the System.Web.Services.Discovery.DiscoveryClientResult.Filename property
:	23	to filename. Name of the type for a reference in the discovery document. Sets the
:	24	System. We b. Services. Discovery. Discovery Client Result. Reference Type Name
:	25	property. URL for the reference. Sets the

1	System.Web.Services.Discovery.DiscoveryClientResult.Url property. Name of
2	the file in which the reference was saved. Sets the
3	System.Web.Services.Discovery.DiscoveryClientResult.Filename property.
4	Filename
5	ToString
6	
7	[C#] public string Filename {get; set;}
8	[C++] public:property String* get_Filename();public:property void
9	set_Filename(String*);
10	[VB] Public Property Filename As String
11	[JScript] public function get Filename(): String; public function set
12	Filename(String);
13	
14	Description
15	Gets or sets the name of the file in which the reference is saved.
16	The name of the file is the name of the file saved on the local machine
17	based on the contents of the file. For instance, if the referenced file is a discovery
18	document, the saved file will have a .disco extension on a Windows operating
19	system.
20	ReferenceTypeName
21	ToString
22	
23	[C#] public string ReferenceTypeName {get; set;}
24	[C++] public:property String* get_ReferenceTypeName();public:property
25	<pre>void set_ReferenceTypeName(String*);</pre>

	1	The URL can refer to a discovery document, XML Schema Definition
	2	(XSD) schema, or a Service Description.
	3	DiscoveryClientResultCollection class (System.Web.Services.Discovery)
	4	ToString
	5	
	6	
	7	Description
	8	Contains a collection of
	9	System.Web.Services.Discovery.DiscoveryClientResult objects. This class
	10	cannot be inherited.
San Seat State Sta	11	DiscoveryClientResultCollection
	12	Example Syntax:
	13	ToString
	14	
	15	[C#] public DiscoveryClientResultCollection();
	16	[C++] public: DiscoveryClientResultCollection();
i.	17	[VB] Public Sub New()
	18	[JScript] public function DiscoveryClientResultCollection();
	19	Count
	20	InnerList
	21	Item
	22	ToString
	23	
	24	
	25	Description

The state that their thair that their their

1

2

3

5

7

8

10

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets the **System.Web.Services.Discovery.DiscoveryClientResult** at position i of the

 $System. Web. Services. Discovery. Discovery Client Result Collection \ .$ 

This property provides the ability to access a specific element in the collection by using the following syntax: myCollection[index]. The zero-based index of the **System.Web.Services.Discovery.DiscoveryClientResult** to get or set.

List

Add

[C#] public int Add(DiscoveryClientResult value);

[C++] public: int Add(DiscoveryClientResult\* value);

[VB] Public Function Add(ByVal value As DiscoveryClientResult) As Integer [JScript] public function Add(value : DiscoveryClientResult) : int;

Description

 $Adds\ a\ System. We b. Services. Discovery. Discovery Client Result\ to\ the$   $System. We b. Services. Discovery. Discovery Client Result Collection\ .$ 

Return Value: The position into which the

System.Web.Services.Discovery.DiscoveryClientResult was inserted. The System.Web.Services.Discovery.DiscoveryClientResult to add to the System.Web.Services.Discovery.DiscoveryClientResultCollection.

Contains

[C#] public bool Contains(DiscoveryClientResult value);

	1	i
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
Maria Maria	11	
	12	
	13	
	14	
nd.	15	
	16	
an and page	17	
	18	
	19	
	20	I

[C++] public: bool Contains(DiscoveryClientResult* value);
[VB] Public Function Contains(ByVal value As DiscoveryClientResult) As
Boolean
[JScript] public function Contains(value : DiscoveryClientResult) : Boolean;
Description
Determines whether the
System.Web.Services.Discovery.DiscoveryClientResultCollection contains a
$specific \ \textbf{System.Web.Services.Discovery.DiscoveryClientResult} \ .$
Return Value: true if the
System.Web.Services.Discovery.DiscoveryClientResult is found in the
${\bf System. Web. Services. Discovery. Discovery Client Result Collection}\ ; \ otherwise,$
${\bf false}\;.\; {\bf The}\; {\bf System. Web. Services. Discovery. Discovery Client Result}\; {\bf to}\; {\bf locate}\; {\bf in}\;$
$the \ System. We b. Services. Discovery. Discovery Client Result Collection.$
Remove
[C#] public void Remove(DiscoveryClientResult value);
[C++] public: void Remove(DiscoveryClientResult* value);
[VB] Public Sub Remove(ByVal value As DiscoveryClientResult)
[JScript] public function Remove(value : DiscoveryClientResult);
Description
Removes the first occurrence of a specific
System.Web.Services.Discovery.DiscoveryClientResult from the
System.Web.Services.Discovery.DiscoveryClientResultCollection

3

5

6

8

9

10

11

12

13

16

17

18

19

20

21

22

23

24

25

The System.Web.Services.Discovery.DiscoveryClientResult elements that follow the removed System.Web.Services.Discovery.DiscoveryClientResult move up to occupy the vacated spot. The

System.Web.Services.Discovery.DiscoveryClientResult to remove from the System.Web.Services.Discovery.DiscoveryClientResultCollection.

DiscoveryClientProtocol.DiscoveryClientResultsFile class (System.Web.Services.Discovery)

**ToString** 

## Description

Represents the root element of an XML document containing the results of all files written when the

System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.String,System.String) method is invoked.

When you invoke the

System.Web.Services.Discovery.DiscoveryClientProtocol.WriteAll(System.String,System.String) method, all resolved discovery documents and a file containing a map of all those files are saved to a directory. The map file is described in XML with the root element being

System.Web.Services.Discovery.DiscoveryClientProtocol.DiscoveryClientRes ultsFile; this class is passed to the System.Xml.Serialization.XmlSerializer class to serialize the results.

DiscoveryClientProtocol.DiscoveryClientResultsFile Example Syntax:

1	ToString
2	
3	[C#] public DiscoveryClientProtocol.DiscoveryClientResultsFile();
4	[C++] public: DiscoveryClientResultsFile();
5	[VB] Public Sub New()
6	[JScript] public function DiscoveryClientProtocol.DiscoveryClientResultsFile();
7	Results
8	ToString
9	
10	[C#] public DiscoveryClientResultCollection Results {get;}
11	[C++] public:property DiscoveryClientResultCollection* get_Results();
12	[VB] Public ReadOnly Property Results As DiscoveryClientResultCollection
13	[JScript] public function get Results(): DiscoveryClientResultCollection;
14	
15	Description
16	Gets a collection of
17	System.Web.Services.Discovery.DiscoveryClientResult objects.
18	DiscoveryDocument class (System.Web.Services.Discovery)
19	ToString
20	
21	
22	Description
23	Represents a discovery document. This class cannot be inherited.
24	Web Services discovery involves discovering the available Web Services
25	given an URL. The URL typically points to a discovery document, which usually

has a .disco file extension. The discovery document, which is an XML document, contains references to information about the existance of Web Services, such as a Service Description, XML Schema Definition language (XSD) schema, or another discovery document. This class represents the contents of the discovery document; where the System. Web. Services. Discovery. Discovery Document. References property contains a list of the references contained within the discovery document.

**ToString** 

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

2

3

4

5

6

7

[C#] public const string Namespace;

[VB] Public Const Namespace As String

[C++] public: const String\* Namespace;

[JScript] public var Namespace : String;

Description

Namespace of the discovery XML element of a discovery document.

A discovery document contains references to documents describing Web Services. These references are XML elements contained within a discovery XML element. That discovery XML element is a member of the XML namespace specified by the

System.Web.Services.Discovery.DiscoveryDocument.Namespace constant.

DiscoveryDocument

Example Syntax:

**ToString** 

[C#] public DiscoveryDocument();

1	[C++] public: DiscoveryDocument();
2	[VB] Public Sub New()
3	[JScript] public function DiscoveryDocument();
4	
5	Description
6	Initializes a new instance of the
7	System.Web.Services.Discovery.DiscoveryDocument class.
8	References
9	ToString
10	
11	[C#] public IList References {get;}
12	[C++] public:property IList* get_References();
13	[VB] Public ReadOnly Property References As IList
14	[JScript] public function get References(): IList;
15	
16	Description
17	A list of references contained within the discovery document.
18	A discovery document contains references to information about the
19	existence of Web Services. These references can refer to Service Descriptions
20	XSD schemas, or other discovery documents. The
21	System.Web.Services.Discovery.DiscoveryDocument.References property
22	contains a list of those references.
23	CanRead
24	
25	[C#] public static bool CanRead(XmlReader xmlReader);

1	[C++] public: static bool CanRead(XmlReader* xmlReader);
2	[VB] Public Shared Function CanRead(ByVal xmlReader As XmlReader) As
3	Boolean
4	[JScript] public static function CanRead(xmlReader : XmlReader) : Boolean;
5	
6	Description
7	Returns a value indicating whether the passed System.Xml.XmlReader
8	can be descrialized into a System. Web. Services. Discovery. Discovery Document
9	
10	Return Value: true if System.Xml.XmlReader can be deserialized into a
11	System.Web.Services.Discovery.DiscoveryDocument; otherwise, false. The
12	System.Xml.XmlReader to determine whether it can be deserialized into a
13	System.Web.Services.Discovery.DiscoveryDocument.
14	Read
15	
16	[C#] public static DiscoveryDocument Read(Stream stream);
17	[C++] public: static DiscoveryDocument* Read(Stream* stream);
18	[VB] Public Shared Function Read(ByVal stream As Stream) As
19	DiscoveryDocument
20	[JScript] public static function Read(stream : Stream) : DiscoveryDocument;
21	Reads and returns a System. Web. Services. Discovery. Discovery Document from
22	the passed object.
23	
24	Description
25	

1	Reads and returns a
2	System.Web.Services.Discovery.DiscoveryDocument from the passed
3	System.IO.Stream.
4	Return Value: A System. Web. Services. Discovery. Discovery Document
5	containing the contents of a discovery document from the passed
6	System.IO.Stream . The System.IO.Stream from which to read the
7	System.Web.Services.Discovery.DiscoveryDocument.
8	Read
9	
10	[C#] public static DiscoveryDocument Read(TextReader reader);
11	[C++] public: static DiscoveryDocument* Read(TextReader* reader);
12	[VB] Public Shared Function Read(ByVal reader As TextReader) As
13	DiscoveryDocument
14	[JScript] public static function Read(reader : TextReader) : DiscoveryDocument;
15	
16	Description
17	Reads and returns a
18	System.Web.Services.Discovery.DiscoveryDocument from the passed
19	System.IO.TextReader.
20	Return Value: A System.Web.Services.Discovery.DiscoveryDocument
21	containing the contents of a discovery document from the passed
22	System.IO.TextReader . The System.IO.TextReader from which to read the
23	System.Web.Services.Discovery.DiscoveryDocument.
24	Read
25	

[C#] public static DiscoveryDocument Read(XmlReader xmlReader); 2 [C++] public: static DiscoveryDocument\* Read(XmlReader\* xmlReader); 3 [VB] Public Shared Function Read(ByVal xmlReader As XmlReader) As 4 DiscoveryDocument 5 [JScript] public static function Read(xmlReader : XmlReader) : 6 DiscoveryDocument; 7 8 Description 9 Reads and returns a 10 System.Web.Services.Discovery.DiscoveryDocument from the passed 11 System.Xml.XmlReader. 12 Return Value: A System. Web. Services. Discovery. Discovery Document 13 containing the contents of a discovery document from the passed 14 System.Xml.XmlReader. The System.Xml.XmlReader from which to read the 15 System.Web.Services.Discovery.DiscoveryDocument. 16 Write 17 18 [C#] public void Write(Stream stream); 19 [C++] public: void Write(Stream\* stream); 20 [VB] Public Sub Write(ByVal stream As Stream) 21 [JScript] public function Write(stream : Stream); 22 23 Description 24

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Writes this System. Web. Services. Discovery. Discovery Document into the passed System.IO.Stream. The System.IO.Stream into which this System.Web.Services.Discovery.DiscoveryDocument is written. Write [C#] public void Write(TextWriter writer); [C++] public: void Write(TextWriter\* writer); [VB] Public Sub Write(ByVal writer As TextWriter) [JScript] public function Write(writer: TextWriter); Writes this System. Web. Services. Discovery. Discovery Document into the passed object. Description Writes this System. Web. Services. Discovery. Discovery Document into the passed System.IO.TextWriter. The System.IO.TextWriter into which this System.Web.Services.Discovery.DiscoveryDocument is written. Write [C#] public void Write(XmlWriter writer); [C++] public: void Write(XmlWriter\* writer); [VB] Public Sub Write(ByVal writer As XmlWriter) [JScript] public function Write(writer: XmlWriter); Description

Writes this System. Web. Services. Discovery. Discovery Document into the
passed System.Xml.XmlWriter . The System.Xml.XmlWriter into which this
System.Web.Services.Discovery.DiscoveryDocument is written.
DiscoveryDocumentLinksPattern class (System.Web.Services.Discovery)
Write
Description
DiscoveryDocumentLinksPattern
Example Syntax:
Write
[C#] public DiscoveryDocumentLinksPattern();
[C++] public: DiscoveryDocumentLinksPattern();
[VB] Public Sub New()
[JScript] public function DiscoveryDocumentLinksPattern();
Pattern
Write
[C#] public override string Pattern {get;}
[C++] public:property virtual String* get_Pattern();
[VB] Overrides Public ReadOnly Property Pattern As String
[JScript] public function get Pattern(): String;

Description

1

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

GetDiscoveryReference

[C#] public override DiscoveryReference GetDiscoveryReference(string filename);

[C++] public: DiscoveryReference\* GetDiscoveryReference(String\* filename);

[VB] Overrides Public Function GetDiscoveryReference(ByVal filename As

String) As DiscoveryReference

[JScript] public override function GetDiscoveryReference(filename : String) :

DiscoveryReference;

Description

DiscoveryDocumentReference class (System.Web.Services.Discovery) **ToString** 

Description

Represents a reference to a discovery document. This class cannot be inherited.

Web Services discovery involves discovering the available Web Services given an URL. The URL typically points to a discovery document, which usually has a .disco file extension. The discovery document contains references to

1	information about the existance of Web Services, such as Service Descriptions,
2	XML Schema Definition language (XSD) schemas, or other discovery documents.
3	This class represents a reference to a discovery document.
4	DiscoveryDocumentReference
5	Example Syntax:
6	ToString
7	
8	[C#] public DiscoveryDocumentReference();
9	[C++] public: DiscoveryDocumentReference();
10	[VB] Public Sub New()
11	[JScript] public function DiscoveryDocumentReference(); Initializes a new
12	instance of the System. Web. Services. Discovery. Discovery Document Reference
13	class.
14	
15	Description
16	Initializes a new instance of the
17	System.Web.Services.Discovery.DiscoveryDocumentReference class.
18	DiscoveryDocumentReference
19	Example Syntax:
20	ToString
21	
22	[C#] public DiscoveryDocumentReference(string href);
23	[C++] public: DiscoveryDocumentReference(String* href);
24	[VB] Public Sub New(ByVal href As String)
25	[JScript] public function DiscoveryDocumentReference(href: String);

T)			
Desci	rin	tin	V
レしいし	$\iota \rho$	$\iota\iota\iota\upsilon$	•

Initializes a new instance of the

System.Web.Services.Discovery.DiscoveryDocumentReference class, setting the System.Web.Services.Discovery.DiscoveryDocumentReference.Ref property to *href*. Reference to a discovery document.

ClientProtocol

DefaultFilename

**ToString** 

## Description

Gets the name of the default file to use when saving the referenced discovery document.

Document

**ToString** 

[C#] public DiscoveryDocument Document {get;}

[C++] public: \_\_property DiscoveryDocument\* get\_Document();

[VB] Public ReadOnly Property Document As DiscoveryDocument

[JScript] public function get Document(): DiscoveryDocument;

## Description

23

24

Gets the contents of the referenced discovery document as a System.Web.Services.Discovery.DiscoveryDocument object.

lee@hayes plic 509-324-9256

1	If the discovery document has not been downloaded and added to the
2	System.Web.Services.Discovery.DiscoveryClientProtocol.Documents property
3	of System.Web.Services.Discovery.DiscoveryReference.ClientProtocol, an
4	attempt to download and resolve the document is made.
5	Ref
6	ToString
7	
8	[C#] public string Ref {get; set;}
9	[C++] public:property String* get_Ref();public:property void
10	set_Ref(String*);
11	[VB] Public Property Ref As String
12	[JScript] public function get Ref(): String; public function set Ref(String);
13	
14	Description
15	Gets or sets the reference to a discovery document.
16	For most references, the
17	System.Web.Services.Discovery.DiscoveryDocumentReference.Ref property is
18	a URL, such as http://www.contoso.com/MyWebService.disco.
19	Url
20	ToString
21	
22	[C#] public override string Url {get; set;}
23	[C++] public:property virtual String* get_Url();public:property virtual void
24	set_Url(String*);
25	[VB] Overrides Public Property Url As String

757 MS1-863US.APP

[JScript] public function get Url(): String; public function set Url(String); 2 Description 3 Gets or sets the URL of the referenced discovery document. Returns the value of the 5  $System. Web. Services. Discovery. Discovery Document Reference. Ref \ property.$ 6 ReadDocument 7 8 [C#] public override object ReadDocument(Stream stream); 9 [C++] public: Object\* ReadDocument(Stream\* stream); 10 [VB] Overrides Public Function ReadDocument(ByVal stream As Stream) As 11 Object 12 [JScript] public override function ReadDocument(stream : Stream) : Object; 13 14 Description 15 Reads and returns the discovery document from the passed 16 System.IO.Stream. 17 Return Value: A System. Web. Services. Discovery. Discovery Document 18 containing the contents of the referenced discovery document. System.IO.Stream 19 containing the discovery document. 20 Resolve 21 22 [C#] protected internal override void Resolve(string contentType, Stream stream); 23 [C++] protected public: void Resolve(String\* contentType, Stream\* stream); 24 [VB] Overrides Protected Friend Dim Sub Resolve(ByVal contentType As String,

ByVal stream As Stream) [JScript] package override function Resolve(contentType : String, stream : 2 Stream); 3 Description 5 Resolves whether the referenced document is valid. 6 If the MIME type is text/xml and the contents of *stream* is a discovery 7 document, then the contents of stream are added to the 8 System.Web.Services.Discovery.DiscoveryClientProtocol.References and 9 System.Web.Services.Discovery.DiscoveryClientProtocol.Documents 10 properties of 11 System.Web.Services.Discovery.DiscoveryReference.ClientProtocol . The 12 MIME type of stream. The System.IO.Stream containing the referenced 13 document. 14 ResolveAll 15 16 [C#] public void ResolveAll(); 17 [C++] public: void ResolveAll(); 18 [VB] Public Sub ResolveAll() 19 [JScript] public function ResolveAll(); 20 21 Description 22 Verifies that all referenced documents within the discovery document are 23 valid. 24 25

Unless you specifically need to resolve the individual references of a System.Web.Services.Discovery.DiscoveryDocumentReference object, you should invoke the

System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveAll or System.Web.Services.Discovery.DiscoveryClientProtocol.ResolveOneLevel methods of System.Web.Services.Discovery.DiscoveryClientProtocol.

WriteDocument

[C#] public override void WriteDocument(object document, Stream stream);

[C++] public: void WriteDocument(Object\* document, Stream\* stream);

[VB] Overrides Public Sub WriteDocument(ByVal document As Object, ByVal stream As Stream)

[JScript] public override function WriteDocument(document : Object, stream : Stream);

Description

Writes the passed  ${\bf System.Web.Services.Discovery.DiscoveryDocument}$  into the passed  ${\bf System.IO.Stream}$  . The

System.Web.Services.Discovery.DiscoveryDocument to write into *stream*. The System.IO.Stream into which the serialized discovery document is written.

DiscoveryDocumentSearchPattern class (System.Web.Services.Discovery)
WriteDocument

Description

1	
2	DiscoveryDocumentSearchPattern
3	Example Syntax:
4	WriteDocument
5	
6	[C#] public DiscoveryDocumentSearchPattern();
7	[C++] public: DiscoveryDocumentSearchPattern();
8	[VB] Public Sub New()
9	[JScript] public function DiscoveryDocumentSearchPattern();
10	Pattern
11	WriteDocument
12	
13	[C#] public override string Pattern {get;}
14	[C++] public:property virtual String* get_Pattern();
15	[VB] Overrides Public ReadOnly Property Pattern As String
16	[JScript] public function get Pattern(): String;
17	
18	Description
19	
20	GetDiscoveryReference
21	
22	[C#] public override DiscoveryReference GetDiscoveryReference(string
23	filename);
24	[C++] public: DiscoveryReference* GetDiscoveryReference(String* filename);
25	[VB] Overrides Public Function GetDiscoveryReference(ByVal filename As

1	String) As DiscoveryReference
2	[JScript] public override function GetDiscoveryReference(filename : String) :
3	DiscoveryReference;
4	
5	Description
6	
7	DiscoveryExceptionDictionary class (System.Web.Services.Discovery)
8	ToString
9	
10	
11	Description
12	Collects exceptions that occurred during Web Services discovery. This
13	class cannot be inherited.
14	The System.Web.Services.Discovery.DiscoveryClientProtocol.Errors
15	property of System. Web. Services. Discovery. Discovery Client Protocol is of type
16	System.Web.Services.Discovery.DiscoveryExceptionDictionary.
17	DiscoveryExceptionDictionary
18	Example Syntax:
19	ToString
20	
21	[C#] public DiscoveryExceptionDictionary();
22	[C++] public: DiscoveryExceptionDictionary();
23	[VB] Public Sub New()
24	[JScript] public function DiscoveryExceptionDictionary();
25	Count

Dictionary InnerHashtable Item **ToString** 6 Description 7 Gets or sets the System. Exception that occurred while discovering the 8 specified URL from the 9 System.Web.Services.Discovery.DiscoveryExceptionDictionary . The URL of 10 the discovery document that caused an exception to be thrown during Web 11 Services discovery. 12 Keys 13 **ToString** 14 15 [C#] public ICollection Keys {get;} 16 [C++] public: property ICollection\* get Keys(); 17 [VB] Public ReadOnly Property Keys As ICollection 18 [JScript] public function get Keys(): ICollection; 19 20 Description 21 Gets a System. Collections. I Collection object with all of the keys in the 22 System.Web.Services.Discovery.DiscoveryExceptionDictionary . 23 Values 24

**ToString** 

1 [C#] public ICollection Values {get;} [C++] public: property ICollection\* get Values(); 3 [VB] Public ReadOnly Property Values As ICollection [JScript] public function get Values(): ICollection; 5 6 Description 7 Gets a System. Collections. I Collection object containing all of the values 8 in the System. Web. Services. Discovery. Discovery Exception Dictionary. 9 Add 10 11 [C#] public void Add(string url, Exception value); 12 [C++] public: void Add(String\* url, Exception\* value); 13 [VB] Public Sub Add(ByVal url As String, ByVal value As Exception) 14 [JScript] public function Add(url : String, value : Exception); 15 16 Description 17 Adds an **System.Exception** with a key of *url* to the 18  $\textbf{System.Web.Services.D} is covery. \textbf{D} is covery. \textbf{ExceptionD} ictionary \ . \ The \ URL \ that$ 19 caused an exception during Web Services discovery. The System. Exception that 20 occurred during Web Services discovery. 21 Contains 22 23 [C#] public bool Contains(string url); 24 [C++] public: bool Contains(String\* url); 25

1	[VB] Public Function Contains(ByVal url As String) As Boolean
2	[JScript] public function Contains(url : String) : Boolean;
3	
4	Description
5	Determines whether the
6	System.Web.Services.Discovery.DiscoveryExceptionDictionary contains an
7	System.Exception with the specified URL.
8	Return Value: true if the
9	System.Web.Services.Discovery.DiscoveryExceptionDictionary contains an
10	System.Exception with the specified URL; otherwise, false. The URL of the
11	System.Exception to locate within the
12	System.Web.Services.Discovery.DiscoveryExceptionDictionary.
13	Remove
14	
15	[C#] public void Remove(string url);
16	[C++] public: void Remove(String* url);
17	[VB] Public Sub Remove(ByVal url As String)
18	[JScript] public function Remove(url : String);
19	
20	Description
21	Removes an System.Exception with the specified URL from the
22	System.Web.Services.Discovery.DiscoveryExceptionDictionary . The URL of
23	the System.Exception to remove from the
24	System.Web.Services.Discovery.DiscoveryExceptionDictionary.
25	DiscoveryReference class (System.Web.Services.Discovery)

1	ToString
2	
3	
4	Description
5	The base class for discoverable references using Web Services discovery.
6	System.Web.Services.Discovery.ContractReference,
7	System.Web.Services.Discovery.SchemaReference, and
8	System.Web.Services.Discovery.DiscoveryDocumentReference all inherit from
9	System.Web.Services.Discovery.DiscoveryReference, and represent the three
10	types of documents disoverable through Web Services discovery: Service
11	Descriptions, XML Schema Definition (XSD) schemas, and discovery documents
12	respectively.
13	DiscoveryReference
14	Example Syntax:
15	ToString
16	
17	[C#] protected DiscoveryReference();
18	[C++] protected: DiscoveryReference();
19	[VB] Protected Sub New()
20	[JScript] protected function DiscoveryReference();
21	ClientProtocol
22	ToString
23	
24	[C#] public DiscoveryClientProtocol ClientProtocol {get; set;}
25	[C++] public:property DiscoveryClientProtocol* get_ClientProtocol();public:

1	property void set_ClientProtocol(DiscoveryClientProtocol*);
2	[VB] Public Property ClientProtocol As DiscoveryClientProtocol
3	[JScript] public function get ClientProtocol() : DiscoveryClientProtocol;public
4	function set ClientProtocol(DiscoveryClientProtocol);
5	
6	Description
7	Gets or sets the instance of
8	System.Web.Services.Discovery.DiscoveryClientProtocol used in a discovery
9	process.
10	DefaultFilename
11	ToString
12	
13	[C#] public virtual string DefaultFilename {get;}
14	[C++] public:property virtual String* get_DefaultFilename();
15	[VB] Overridable Public ReadOnly Property DefaultFilename As String
16	[JScript] public function get DefaultFilename() : String;
17	
18	Description
19	Gets the name of the default file to use when saving the referenced
20	discovery document, XSD schema, or Service Description.
21	Url
22	ToString
23	
24	[C#] public abstract string Url {get; set;}
25	[C++] public:property virtual String* get_Url() = 0;public:property virtual

void set\_Url(String\*) = 0; [VB] MustOverride Public Property Url As String 2 [JScript] public abstract function get Url(): String; public abstract function set 3 Url(String); 5 Description 6 Gets or sets the URL of the referenced document. 7 FilenameFromUrl 8 9 [C#] protected static string FilenameFromUrl(string url); 10 [C++] protected: static String\* FilenameFromUrl(String\* url); 11 [VB] Protected Shared Function FilenameFromUrl(ByVal url As String) As String 12 [JScript] protected static function FilenameFromUrl(url: String): String; 13 14 Description 15 Returns a file name based on the passed URL. 16 Return Value: Name of the file based on the passed URL. The URL on which the 17 name of the file is based. 18 ReadDocument 19 20 [C#] public abstract object ReadDocument(Stream stream); 21 [C++] public: virtual Object\* ReadDocument(Stream\* stream) = 0; 22 [VB] MustOverride Public Function ReadDocument(ByVal stream As Stream) As 23 Object 24 [JScript] public abstract function ReadDocument(stream : Stream) : Object; 25

Description

Reads the passed **System.IO.Stream** and returns an instance of the class representing the type of referenced document.

*Return Value*: An **System.Object** with an underlying type matching the type of referenced document.

The return value is always an **System.Object**. However, the underlying type varies, depending on the type of document referenced. For instance, a **System.IO.Stream** passed into

System.Web.Services.Discovery.SchemaReference.ReadDocument(System.IO .Stream) returns an instance of System.Xml.Schema.XmlSchema .

System.IO.Streamcontaining the reference document.

Resolve

[C#] public void Resolve();

[C++] public: void Resolve();

[VB] Public Sub Resolve()

[JScript] public function Resolve(); Resolves whether the referenced document is valid.

Description

Downloads the referenced document at

**System.Web.Services.Discovery.DiscoveryReference.Url** to resolve whether the referenced document is valid.

Downloads the referenced document and then invokes the overloaded System.Web.Services.Discovery.DiscoveryReference.Resolve member that takes a contentType and a System.IO.Stream. If the MIME type and the contents of stream match those expected by the reference type, then the contents of stream are added to the

System.Web.Services.Discovery.DiscoveryClientProtocol.References and System.Web.Services.Discovery.DiscoveryClientProtocol.Documents properties of

 $System. Web. Services. Discovery. Discovery Reference. Client Protocol\ .$ 

Resolve

[C#] protected internal abstract void Resolve(string contentType, Stream stream);[C++] protected public: virtual void Resolve(String\* contentType, Stream\*stream) = 0;

[VB] MustOverride Protected Friend Dim Sub Resolve(ByVal contentType As String, ByVal stream As Stream)

[JScript] package abstract function Resolve(contentType : String, stream : Stream);

## Description

Resolves whether the referenced document is valid.

If the MIME type and the contents of *stream* match those expected by the reference type, then the contents of *stream* are added to the

System.Web.Services.Discovery.DiscoveryClientProtocol.References and System.Web.Services.Discovery.DiscoveryClientProtocol.Documents

properties of  $System. Web. Services. Discovery. Discovery Reference. Client Protocol\ .\ The$ 2 MIME type of stream. The System.IO.Stream containing the referenced 3 document. WriteDocument 5 6 [C#] public abstract void WriteDocument(object document, Stream stream); 7 [C++] public: virtual void WriteDocument(Object\* document, Stream\* stream) = 8 0; 9 [VB] MustOverride Public Sub WriteDocument(ByVal document As Object, 10 ByVal stream As Stream) 11 [JScript] public abstract function WriteDocument(document : Object, stream : 12 Stream); 13 14 Description 15 When overridden in a derived class, writes the document into a 16 System.IO.Stream. The document to write into a System.IO.Stream. The 17 System.IO.Stream into which the document is written. 18 DiscoveryReferenceCollection class (System.Web.Services.Discovery) 19 WriteDocument 20 21 22 Description 23 A collection of discovery references. This class cannot be inherited. 24 DiscoveryReferenceCollection 25

Example Syntax: WriteDocument 2 3 [C#] public DiscoveryReferenceCollection(); [C++] public: DiscoveryReferenceCollection(); 5 [VB] Public Sub New() 6 [JScript] public function DiscoveryReferenceCollection(); 7 Count 8 InnerList Item 10 WriteDocument 11 12 13 Description 14 Gets or sets the System. Web. Services. Discovery. Discovery Reference at 15 the specified index. The zero-based index of the 16 System.Web.Services.Discovery.DiscoveryReference to get or set. 17 List 18 Add 19 20 [C#] public int Add(DiscoveryReference value); 21 [C++] public: int Add(DiscoveryReference\* value); 22 [VB] Public Function Add(ByVal value As DiscoveryReference) As Integer 23 [JScript] public function Add(value : DiscoveryReference) : int; 24 25

15

16

17

18

19

20

21

22

23

24

5	
6	
7	
8	
9	
10	
11	
12	
13	

2

3

Description	1
-------------	---

 ${\bf Adds~a~System. Web. Services. Discovery. Discovery Reference~to~the} \\ {\bf System. Web. Services. Discovery. Discovery Reference Collection~.}$ 

Return Value: The position into which the

System.Web.Services.Discovery.DiscoveryReference was inserted into the System.Web.Services.Discovery.DiscoveryReferenceCollection . The System.Web.Services.Discovery.DiscoveryReference to add to the System.Web.Services.Discovery.DiscoveryReferenceCollection.

**Contains** 

[C#] public bool Contains(DiscoveryReference value);

[C++] public: bool Contains(DiscoveryReference\* value);

[VB] Public Function Contains(ByVal value As DiscoveryReference) As Boolean

[JScript] public function Contains(value : DiscoveryReference) : Boolean;

### Description

Determines whether the

System.Web.Services.Discovery.DiscoveryReferenceCollection contains a specific System.Web.Services.Discovery.DiscoveryReference.

Return Value: true if the

System.Web.Services.Discovery.DiscoveryReferenceCollection contains the

 ${\bf System. Web. Services. Discovery. Discovery Reference}\ ;\ otherwise,\ {\bf false}\ .\ The$ 

System.Web.Services.Discovery.DiscoveryReference to locate within the

System. Web. Services. Discovery. Discovery Reference Collection.

	,:	1500
	١,	<u>.</u>
	i,	
	ě,	ij
	1	## j
	1	Harrie I
	ľ,	
	ť,	4
新作""是"有"等等等	Ļ	
	ť,	estil estil
	ì,	
	;	::::::::::::::::::::::::::::::::::::::
l sáb		
	;	;sita

1	Remove
2	
3	[C#] public void Remove(DiscoveryReference value);
4	[C++] public: void Remove(DiscoveryReference* value);
5	[VB] Public Sub Remove(ByVal value As DiscoveryReference)
6	[JScript] public function Remove(value : DiscoveryReference);
7	
8	Description
9	Removes a System. Web. Services. Discovery. Discovery Reference from
10	the System.Web.Services.Discovery.DiscoveryReferenceCollection . The
11	System.Web.Services.Discovery.DiscoveryReference to remove from the
12	System.Web.Services.Discovery.DiscoveryReferenceCollection.
13	DiscoveryRequestHandler class (System.Web.Services.Discovery)
14	ToString
15	
16	
17	Description
18	
19	DiscoveryRequestHandler
20	Example Syntax:
21	ToString
22	
23	[C#] public DiscoveryRequestHandler();
24	[C++] public: DiscoveryRequestHandler();
25	

,	[VB] Public Sub New()
1	
2	[JScript] public function DiscoveryRequestHandler();
3	IsReusable
4	ToString
5	
6	[C#] public bool IsReusable {get;}
7	[C++] public:property bool get_IsReusable();
8	[VB] Public ReadOnly Property IsReusable As Boolean
9	[JScript] public function get IsReusable(): Boolean;
10	
11	Description
12	
13	ProcessRequest
14	
15	[C#] public void ProcessRequest(HttpContext context);
16	[C++] public:sealed void ProcessRequest(HttpContext* context);
17	[VB] NotOverridable Public Sub ProcessRequest(ByVal context As HttpContext)
18	[JScript] public function ProcessRequest(context : HttpContext);
19	
20	Description
21	
22	DiscoverySearchPattern class (System.Web.Services.Discovery)
23	ToString
24	
25	

	,	
	2	
	3	Description
	4	
	5	DiscoverySearchPattern
	6	Example Syntax:
	7	ToString
	8	
	9	[C#] protected DiscoverySearchPattern();
	10	[C++] protected: DiscoverySearchPattern();
the transfer of the transfer o	11	[VB] Protected Sub New()
	12	[JScript] protected function DiscoverySearchPattern();
	13	Pattern
	14	ToString
# # #	15	
the think think the three think	16	[C#] public abstract string Pattern {get;}
**	17	[C++] public:property virtual String* get_Pattern() = 0;
	18	[VB] MustOverride Public ReadOnly Property Pattern As String
	19	[JScript] public abstract function get Pattern() : String;
	20	
	21	Description
	22	
	23	GetDiscoveryReference
	24	
	25	[C#] public abstract DiscoveryReference GetDiscoveryReference(string filename);
	23	E-max

	1	[C++] public: virtual DiscoveryReference* GetDiscoveryReference(String*
	2	filename = 0;
	3	[VB] MustOverride Public Function GetDiscoveryReference(ByVal filename As
	4	String) As DiscoveryReference
	5	[JScript] public abstract function GetDiscoveryReference(filename : String) :
	6	DiscoveryReference;
	7	
	8	Description
	9	
	10	DynamicDiscoveryDocument class (System.Web.Services.Discovery)
	11	ToString
	12	
	13	
	14	Description
	15	This represents a discovery file.
	16	ToString
ļ: da	17	
	18	[C#] public const string Namespace;
	19	[C++] public: const String* Namespace;
	20	[VB] Public Const Namespace As String
	21	[JScript] public var Namespace : String;
	22	
	23	Description
	24	
	25	DynamicDiscoveryDocument

		Example Syntax:
	2	ToString
	3	
	4	[C#] public DynamicDiscoveryDocument();
	5	[C++] public: DynamicDiscoveryDocument();
	6	[VB] Public Sub New()
	7	[JScript] public function DynamicDiscoveryDocument();
	8	
	9	Description
: 35,	10	Default constructor.
	11	ExcludePaths
	12	ToString
	13	
1	14	[C#] public ExcludePathInfo[] ExcludePaths {get; set;}
1 12 12 12 12 12 12 12 12 12 12 12 12 12	15	[C++] public:property ExcludePathInfo* get_ExcludePaths();public:
	16	property void set_ExcludePaths(ExcludePathInfo*[]);
: =15 : =16	17	[VB] Public Property ExcludePaths As ExcludePathInfo ()
	18	[JScript] public function get ExcludePaths(): ExcludePathInfo[];public function
	19	set ExcludePaths(ExcludePathInfo[]);
	20	
	21	Description
	22	
	23	Load
	24	
	25	[C#] public static DynamicDiscoveryDocument Load(Stream stream);

	1	[C++] public: static DynamicDiscoveryDocument* Load(Stream* stream);
	. 2	[VB] Public Shared Function Load(ByVal stream As Stream) As
	3	DynamicDiscoveryDocument
	4	[JScript] public static function Load(stream : Stream) :
	5	DynamicDiscoveryDocument;
	6	
	7	Description
	8	Read an instance of WebMethodsFile from a stream.
	9	Write
	10	
	11	[C#] public void Write(Stream stream);
	12	[C++] public: void Write(Stream* stream);
	13	[VB] Public Sub Write(ByVal stream As Stream)
	14	[JScript] public function Write(stream : Stream);
and the property of the proper	15	
	16	Description
::::::::::::::::::::::::::::::::::::::	17	Write this instance to a stream.
	18	ExcludePathInfo class (System.Web.Services.Discovery)
	19	Write
	20	
	21	
	22	Description
	23	
	24	ExcludePathInfo
	25	Example Syntax:

```
Write
2
    [C#] public ExcludePathInfo();
    [C++] public: ExcludePathInfo();
4
    [VB] Public Sub New()
    [JScript] public function ExcludePathInfo();
6
7
    Description \\
8
9
           ExcludePathInfo
10
           Example Syntax:
11
           Write
12
13
    [C#] public ExcludePathInfo(string path);
14
    [C++] public: ExcludePathInfo(String* path);
15
    [VB] Public Sub New(ByVal path As String)
16
    [JScript] public function ExcludePathInfo(path : String);
17
18
    Description
19
20
            Path
21
            Write
22
23
     [C#] public string Path {get; set;}
24
     [C++] public: __property String* get_Path();public: __property void
```

19

20

21

22

23

24

set Path(String\*); [VB] Public Property Path As String [JScript] public function get Path(): String; public function set Path(String); 3 4 Description 5 6 SchemaReference class (System.Web.Services.Discovery) 7 **ToString** 8 9 10 Description 11 Represents a reference in a discovery document to an XML Schema 12 Definition language (XSD) schema. This class cannot be inherited. 13 Web Services discovery involves discovering the available Web Services 14 given an URL. A The URL typically points to a discovery document, that usually 15 has a .disco file extension. The discovery document contains references to 16 information about the existance of Web Services. These references can refer to 17

**ToString** 

[C#] public const string Namespace;[C++] public: const String\* Namespace;[VB] Public Const Namespace As String[JScript] public var Namespace: String;

represents a reference to an XSD schema.

Service Descriptions, XSD schemas, or other discovery documents. This class

તી પૈક્ષણ માત્રી તે દેશના પૈકારે માત્રી ઉત્તરી ઉત્તર પૈકારો ઉત્તર પૈકારો પૈકા

D	escri	n	tio	n
IJ	escri	$\nu$	$\iota\iota\upsilon$	"

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

XML namespace for XSD schema references in discovery documents.

Within a discovery document, a reference to an XSD schema is contained within a **schemaRef** XML element, which is a part of the XML namespace specified in the **System.Web.Services.Discovery.SchemaReference.Namespace** constant.

SchemaReference

Example Syntax:

**ToString** 

[C#] public SchemaReference();

[C++] public: SchemaReference();

[VB] Public Sub New()

[JScript] public function SchemaReference(); Initializes a new instance of the

 ${\bf System. Web. Services. Discovery. Schema Reference \ class.}$ 

# Description

Initializes a new instance of the

System. Web. Services. Discovery. Schema Reference class using default values.

SchemaReference

Example Syntax:

**ToString** 

[C#] public SchemaReference(string url);

MS1-863US.APP

	1	[C++] public: SchemaReference(String* url);
	2	[VB] Public Sub New(ByVal url As String)
	3	[JScript] public function SchemaReference(url: String);
	4	
	5	Description
	6	Initializes a new instance of the
	7	System.Web.Services.Discovery.SchemaReference class using the supplied
	8	URL as the XSD schema reference. The URL for the XSD schema. Initializes the
	9	System.Web.Services.Discovery.SchemaReference.Ref property.
	10	ClientProtocol
Tank Tank	11	DefaultFilename
Hom This	12	ToString
Souls shall sport stone stands shall shart shart	13	
	14	
The farm the	15	Description
de source source	16	Gets the name of the default file to use when saving the referenced XSD
j iz	17	schema.
	18	Ref
	19	ToString
	20	
	21	[C#] public string Ref {get; set;}
	22	[C++] public:property String* get_Ref();public:property void
	23	set_Ref(String*);
	24	[VB] Public Property Ref As String
	25	[JScript] public function get Ref(): String; public function set Ref(String);

Description

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the URL to the referenced XSD schema.

In a discovery document, a reference to an XSD schema is contained within a schemaRef XML element. The schemaRef XML element has a ref attribute, which is the URL for the referenced XSD schema. The

**System.Web.Services.Discovery.ContractReference.Ref** property represents the value of the **ref** attribute.

Schema

**ToString** 

[C#] public XmlSchema Schema {get;}

[C++] public: \_\_property XmlSchema\* get\_Schema();

[VB] Public ReadOnly Property Schema As XmlSchema

[JScript] public function get Schema(): XmlSchema;

Description

Gets an **System.Xml.Schema.XmlSchema** object representing the XSD schema.

If the XSD schema has not been downloaded and added to the System.Web.Services.Discovery.DiscoveryClientProtocol.Documents property of System.Web.Services.Discovery.DiscoveryReference.ClientProtocol, an attempt to download and resolve the document is made.

TargetNamespace

**ToString** 

1	
2	[C#] public string TargetNamespace {get; set;}
3	[C++] public:property String* get_TargetNamespace();public:property void
4	set_TargetNamespace(String*);
5	[VB] Public Property TargetNamespace As String
6	[JScript] public function get TargetNamespace(): String;public function set
7	TargetNamespace(String);
8	
9	Description
10	Gets or sets the targetNamespace XML attribute of the XSD schema.
11	Url
12	ToString
13	
14	[C#] public override string Url {get; set;}
15	[C++] public:property virtual String* get_Url();public:property virtual void
16	set_Url(String*);
17	[VB] Overrides Public Property Url As String
18	[JScript] public function get Url(): String; public function set Url(String);
19	
20	Description
21	Gets or sets the URL for the schema reference.
22	For the System. Web. Services. Discovery. Schema Reference class, the
23	System.Web.Services.Discovery.SchemaReference.Url property returns the
24	value of the System. Web. Services. Discovery. Schema Reference. Ref property.
25	ReadDocument

П	
1	
2	[C#] public override object ReadDocument(Stream stream);
3	[C++] public: Object* ReadDocument(Stream* stream);
4	[VB] Overrides Public Function ReadDocument(ByVal stream As Stream) As
5	Object
6	[JScript] public override function ReadDocument(stream : Stream) : Object;
7	
8	Description
9	Reads and returns the XSD schema from the passed System.IO.Stream.
10	Return Value: An System.Xml.Schema.XmlSchema containing the contents of
11	the referenced XSD schema. System.IO.Stream containing the XSD schema.
12	Resolve
13	
14	[C#] protected internal override void Resolve(string contentType, Stream stream);
15	[C++] protected public: void Resolve(String* contentType, Stream* stream);
16	[VB] Overrides Protected Friend Dim Sub Resolve(ByVal contentType As String,
17	ByVal stream As Stream)
18	[JScript] package override function Resolve(contentType : String, stream :
19	Stream);
20	
21	Description
22	Resolves whether the the referenced document is valid.
23	If the MIME type is text/xml and the contents of stream are an XSD
24	schema, then the contents of stream are added to the
25	System.Web.Services.Discovery.DiscoveryClientProtocol.References and

1	System.Web.Services.Discovery.DiscoveryClientProtocol.Documents
2	properties of
3	$System. Web. Services. Discovery. Discovery Reference. Client Protocol\ .\ The$
4	MIME content type of stream. The System.IO.Stream containing the referenced
5	document.
6	WriteDocument
7	
8	[C#] public override void WriteDocument(object document, Stream stream);
9	[C++] public: void WriteDocument(Object* document, Stream* stream);
10	[VB] Overrides Public Sub WriteDocument(ByVal document As Object, ByVal
11	stream As Stream)
12	[JScript] public override function WriteDocument(document : Object, stream :
13	Stream);
14	
15	Description
16	Writes the passed XSD schema into the passed System.IO.Stream. The
17	System.Xml.Schema.XmlSchema to write into stream. The System.IO.Stream
18	into which the serialized XSD schema is written.
19	SoapBinding class (System.Web.Services.Discovery)
20	WriteDocument
21	
22	
23	Description
24	Represents a SOAP binding in a discovery document. This class cannot be
25	inherited.

4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

16

17

18

19

20

21

22

23

24

25

1

3

A SOAP binding is similar to an interface for Web Services and useful for versioning and when a developer is looking for a Web Service implementing a specific SOAP binding. For instance, a discovery document may contain optional information about SOAP Bindings, which specify the SOAP bindings implemented by referenced Web Servcies. SOAP bindings are specified within a discovery document by adding a soap XML element with an XML namespace equal to the System.Web.Services.Discovery.SoapBinding.Namespace constant. The System.Web.Services.Discovery.SoapBinding.Address property specifies the URL to the Web Service and the System.Web.Services.Discovery.SoapBinding.Binding property specifies the SOAP binding implemented by that Web Service.

WriteDocument

[C#] public const string Namespace;

[C++] public: const String\* Namespace;

[VB] Public Const Namespace As String

[JScript] public var Namespace : String;

Description

The XML namespace of the element that specifies a SOAP binding within a discovery document.

SOAP bindings within a discovery document reside within a soap XML element that is a member of the XML namespace specifed by the

 $System. Web. Services. Discovery. Soap Binding. Names pace \ constant.$ 

SoapBinding

1	Example Syntax:
2	WriteDocument
3	
4	[C#] public SoapBinding();
5	[C++] public: SoapBinding();
6	[VB] Public Sub New()
7	[JScript] public function SoapBinding();
8	Address
9	WriteDocument
10	
11	[C#] public string Address {get; set;}
12	[C++] public:property String* get_Address();public:property void
13	set_Address(String*);
14	[VB] Public Property Address As String
15	[JScript] public function get Address() : String;public function set
16	Address(String);
17	
18	Description
19	Gets or sets the URL of the Web Service implementing the SOAP binding.
20	Binding
21	WriteDocument
22	
23	[C#] public XmlQualifiedName Binding {get; set;}
24	[C++] public:property XmlQualifiedName* get_Binding();public:property
25	void set_Binding(XmlQualifiedName*);

1	[VB] Public Property Binding As XmlQualifiedName
2	[JScript] public function get Binding(): XmlQualifiedName; public function set
3	Binding(XmlQualifiedName);
4	
5	Description
6	Gets or sets the XML qualified name of the SOAP binding implemented by
7	the Web Service.
8	XmlSchemaSearchPattern class (System.Web.Services.Discovery)
9	ToString
10	
11	
12	Description
13	
14	XmlSchemaSearchPattern
15	Example Syntax:
16	ToString
17	
18	[C#] public XmlSchemaSearchPattern();
19	[C++] public: XmlSchemaSearchPattern();
20	[VB] Public Sub New()
21	[JScript] public function XmlSchemaSearchPattern();
22	Pattern
23	ToString
24	
25	[C#] public override string Pattern {get;}

[C++] public: \_\_property virtual String\* get\_Pattern(); [VB] Overrides Public ReadOnly Property Pattern As String [JScript] public function get Pattern(): String; 3 4 Description 5 6 7 8 System.Web.Services.Protocols 9 10 Description 11 The System. Web. Services. Protocols namespace consists of the classes 12 that define the protocols used to transmit data across the wire during the 13 communication between ASP.NET Web Service clients and Web Services. 14 AnyReturnReader class (System.Web.Services.Protocols) 15 16 17 Description 18 19 Constructors: 20 AnyReturnReader 21 Example Syntax: 22 23 [C#] public AnyReturnReader(); 24

[C++] public: AnyReturnReader();

	1	[VB] Public Sub New()
	2	[JScript] public function AnyReturnReader();
	3	Methods:
	4	GetInitializer
	5	
	6	[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);
	7	[C++] public: Object* GetInitializer(LogicalMethodInfo* methodInfo);
	8	[VB] Overrides Public Function GetInitializer(ByVal methodInfo As
	9	LogicalMethodInfo) As Object
	10	[JScript] public override function GetInitializer(methodInfo: LogicalMethodInfo)
	11	: Object;
	12	
	13	Description
	14	
	15	Initialize
	16	
	17	[C#] public override void Initialize(object o);
	18	[C++] public: void Initialize(Object* o);
	19	[VB] Overrides Public Sub Initialize(ByVal o As Object)
	20	[JScript] public override function Initialize(o : Object);
	21	
	22	Description
	23	
	24	Read
	25	

	^ II	
	2	[C#] public override object Read(WebResponse response, Stream
	3	responseStream);
	4	[C++] public: Object* Read(WebResponse* response, Stream* responseStream);
	5	[VB] Overrides Public Function Read(ByVal response As WebResponse, ByVal
	6	responseStream As Stream) As Object
	7	[JScript] public override function Read(response: WebResponse, responseStream
	8	: Stream) : Object;
	9	
, Jung	10	Description
ir sant that if the sant sant that that the team that the tark that	11	
	12	HtmlFormParameterReader class (System.Web.Services.Protocols)
	13	ToString
	14	
	15	
	16	Description
	17	
	18	HtmlFormParameterReader
	19	Example Syntax:
	20	ToString
	21	
	22	[C#] public HtmlFormParameterReader();
	23	[C++] public: HtmlFormParameterReader();
	24	[VB] Public Sub New()
	25	[JScript] public function HtmlFormParameterReader();

```
Read
2
    [C#] public override object[] Read(HttpRequest request);
    [C++] public: Object* Read(HttpRequest* request) __gc[];
    [VB] Overrides Public Function Read(ByVal request As HttpRequest) As Object()
    [JScript] public override function Read(request : HttpRequest) : Object[];
    Description
9
           HtmlFormParameterWriter class (System.Web.Services.Protocols)
10
           ToString
11
12
13
    Description
14
15
           HtmlFormParameterWriter
16
           Example Syntax:
17
           ToString
18
19
    [C#] public HtmlFormParameterWriter();
20
    [C++] public: HtmlFormParameterWriter();
21
    [VB] Public Sub New()
22
    [JScript] public function HtmlFormParameterWriter();
23
           Properties:
24
           RequestEncoding
25
```

UsesWriteRequest 1 **ToString** 2 3 Description 5 6 InitializeRequest 7 8 [C#] public override void InitializeRequest(WebRequest request, object[] values); 9 [C++] public: void InitializeRequest(WebRequest\* request, Object\* values 10 \_\_gc[]); 11 [VB] Overrides Public Sub InitializeRequest(ByVal request As WebRequest, 12 ByVal values() As Object) 13 [JScript] public override function InitializeRequest(request : WebRequest, values : 14 Object[]); 15 16 Description 17 18 WriteRequest 19 20 [C#] public override void WriteRequest(Stream requestStream, object[] values); 21 [C++] public: void WriteRequest(Stream\* requestStream, Object\* values \_\_gc[]); 22 [VB] Overrides Public Sub WriteRequest(ByVal requestStream As Stream, ByVal 23 values() As Object) 24 [JScript] public override function WriteRequest(requestStream : Stream, values :

Object[]); 2 Description HttpGetClientProtocol class (System.Web.Services.Protocols) WriteRequest Description Specifies the class for ASP.NET Web Service client proxies that use the 10 HTTP-GET protocol. 11 ASP.NET incorporates two distinct Web Services functionalities: Building 12 ASP.NET Web Services and Building Web Services clients. If you are building a 13 Web Service client using ASP.NET, then a proxy class deriving indirectly or 14 directly from System. Web. Services. Protocols. Web Client Protocol needs to be 15 created for the Web Service you want to call. When the Web Service client calls 16 the Web Service using HTTP, derive the proxy class from 17 System. Web. Services. Protocols. HttpSimpleClientProtocol, which in turn 18 derives from System. Web. Services. Protocols. Web Client Protocol. 19 HttpGetClientProtocol 20 Example Syntax: 21 WriteRequest 22 23 [C#] public HttpGetClientProtocol(); 24 [C++] public: HttpGetClientProtocol();

[VB] Public Sub New() [JScript] public function HttpGetClientProtocol(); 3 Description Initializes a new instance of the  $System. Web. Services. Protocols. Http GetClientProtocol\ class.$ AllowAutoRedirect ClientCertificates ConnectionGroupName Container CookieContainer 11 Credentials DesignMode 13 **Events** 14 PreAuthenticate 15 Proxy 16 RequestEncoding 17 Site 18 Timeout 19 Url 20 UserAgent 21 GetWebRequest 22 23 [C#] protected override WebRequest GetWebRequest(Uri uri); 24 [C++] protected: WebRequest\* GetWebRequest(Uri\* uri);

[VB] Overrides Protected Function GetWebRequest(ByVal uri As Uri) As WebRequest

[JScript] protected override function GetWebRequest(uri : Uri) : WebRequest;

#### Description

4

5

6

7

8

10

11

12

13

15

16

17

18

19

20

21

22

23

24

Creates a System.Net.WebRequest instance for the specified URI.

Return Value: The System.Net.WebRequest instance.

This method overrides the base version of

#### System. Web. Services. Protocols. Web Client Protocol. Get Web Request (System. Web. Services) and the support of the suppor

Uri) to specify that the HTTP request to the Web Service is made using HTTP-GET. By overriding this method, additional customizations can be made to the System.Net.WebRequest object before the Web Service request is made. For example you could add a custom header to the request. The System.Uri to use when creating the System.Net.WebRequest.

HttpMethodAttribute class (System.Web.Services.Protocols)
ToString

#### Description

Applying this attribute to a Web Service client using HTTP-GET or HTTP-POST, sets the types that serialize the parameters sent to a Web Service method and read the response from the Web Service method. This class cannot be inherited.

If a Web Service client invokes a Web Service method using HTTP-GET, System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter must

1	be set to XmlReturnReader and
2	System.Web.Services.Protocols.HttpMethodAttribute.ParameterFormatter
3	set to UrlParameterWriter. Web Service clients invoking a Web Service using
4	HTTP-POST must set
5	System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter to
6	XmlReturnReader and
7	System.Web.Services.Protocols.HttpMethodAttribute.ParameterFormatter to
8	HtmlFormParameterWriter .
9	HttpMethodAttribute
10	Example Syntax:
11	ToString
12	
13	[C#] public HttpMethodAttribute();
14	[C++] public: HttpMethodAttribute();
15	[VB] Public Sub New()
16	[JScript] public function HttpMethodAttribute(); Initializes a new instance of the
17	System.Web.Services.Protocols.HttpMethodAttribute class.
18	
19	Description
20	Initializes a new instance of the
21	System.Web.Services.Protocols.HttpMethodAttribute class.
22	HttpMethodAttribute
23	Example Syntax:
24	ToString
25	

```
[C#] public HttpMethodAttribute(Type returnFormatter, Type
    parameterFormatter);
    [C++] public: HttpMethodAttribute(Type* returnFormatter, Type*
    parameterFormatter);
    [VB] Public Sub New(ByVal returnFormatter As Type, ByVal
    parameterFormatter As Type)
    [JScript] public function HttpMethodAttribute(returnFormatter: Type,
    parameterFormatter: Type);
10
    Description
11
           Initializes a new instance of the
12
    System.Web.Services.Protocols.HttpMethodAttribute . Initializes the
13
    System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter
14
    property to a System. Type that descrializes the response from a Web Service
15
    method. Initializes the
16
    System.Web.Services.Protocols.HttpMethodAttribute.ParameterFormatter
    property to a System. Type that serializes parameters sent from a Web Service
18
    client to a Web Service method.
19
           ParameterFormatter
20
           ToString
21
22
    [C#] public Type ParameterFormatter {get; set;}
23
    [C++] public: __property Type* get ParameterFormatter();public: __property
24
```

void set\_ParameterFormatter(Type\*);

1	[VB] Public Property ParameterFormatter As Type
2	[JScript] public function get ParameterFormatter(): Type;public function set
3	ParameterFormatter(Type);
4	
5	Description
6	Gets or sets a System. Type that serializes parameters sent from a Web
7	Service client to the Web Service method.
8	If the Web Service client is invoking a Web Service method using HTTP-
9	GET or HTTP-POST,
10	System.Web.Services.Protocols.HttpMethodAttribute.ParameterFormatter
11	must be XMLReturnReader.
12	ReturnFormatter
13	ToString
14	
15	[C#] public Type ReturnFormatter {get; set;}
16	[C++] public:property Type* get_ReturnFormatter();public:property void
17	set_ReturnFormatter(Type*);
18	[VB] Public Property ReturnFormatter As Type
19	[JScript] public function get ReturnFormatter(): Type;public function set
20	ReturnFormatter(Type);
21	
22	Description
23	Gets or sets a System. Type that deserializes the response from a Web
24	Service method.
25	

If the Web Service client is invoking a Web Service method using HTTP-GET, System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter must be UrlParameterWriter, whereas a client using HTTP-POST must set System.Web.Services.Protocols.HttpMethodAttribute.ReturnFormatter to HtmlFormParameterWriter.

TypeId

HttpPostClientProtocol class (System.Web.Services.Protocols)

**ToString** 

Description

Specifies the class for ASP.NET Web Service client proxies that use the HTTP-POST protocol.

ASP.NET incorporates two distinct Web Services functionalities: Building ASP.NET Web Services and Building Web Services clients. If you are building a Web Service client using ASP.NET, then a proxy class deriving indirectly or directly from **System.Web.Services.Protocols.WebClientProtocol** needs to be created for the Web Service you want to call. When the Web Service client is calling using HTTP, the proxy class should derive from

System.Web.Services.Protocols.HttpSimpleClientProtocol, which derives from System.Web.Services.Protocols.WebClientProtocol.

HttpPostClientProtocol

Example Syntax:

**ToString** 

```
[C#] public HttpPostClientProtocol();
    [C++] public: HttpPostClientProtocol();
    [VB] Public Sub New()
    [JScript] public function HttpPostClientProtocol();
6
    Description
           Initializes a new instance of the
8
    System. Web. Services. Protocols. HttpPostClientProtocol\ class.
           AllowAutoRedirect
10
           ClientCertificates
11
           ConnectionGroupName
12
           Container
13
           CookieContainer
14
           Credentials
15
           DesignMode
16
           Events
17
           PreAuthenticate
18
           Proxy
19
           RequestEncoding
20
           Site
21
           Timeout
22
           Url
23
           UserAgent
24
           GetWebRequest
25
```

Are in Ary
Ü
ii
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
, <sub>20</sub>
ı silə
: व्यक्ति

1	
2	[C#] protected override WebRequest GetWebRequest(Uri uri);
3	[C++] protected: WebRequest* GetWebRequest(Uri* uri);
4	[VB] Overrides Protected Function GetWebRequest(ByVal uri As Uri) As
5	WebRequest
6	[JScript] protected override function GetWebRequest(uri : Uri) : WebRequest;
7	
8	Description
9	Creates a System.Net.WebRequest instance for the specified URI.
10	Return Value: The System.Net.WebRequest instance.
11	This method overrides the base version of
12	System. We b. Services. Protocols. We bC lient Protocol. Get We bRequest (System.
13	Uri) to specify that the HTTP request to the Web Service is made using HTTP-
14	POST. By overriding this method, you can customize the
15	System.Net.WebRequest object before the Web Service request is made. For
16	example, you can add a custom header to the request. The System.Uri to use when
17	creating the System.Net.WebRequest.
18	HttpSimpleClientProtocol class (System.Web.Services.Protocols)
19	ToString
20	
21	
22	Description
- 11	

The base class for communicating with an Web Service using HTTP-GET and HTTP-POST.

1	Specifies most of the implementation for communicating with an Web
2	Service over HTTP.
3	HttpSimpleClientProtocol
4	Example Syntax:
5	ToString
6	
7	[C#] protected HttpSimpleClientProtocol();
8	[C++] protected: HttpSimpleClientProtocol();
9	[VB] Protected Sub New()
10	[JScript] protected function HttpSimpleClientProtocol();
11	
12	Description
13	Initializes a new instance of the
14	System.Web.Services.Protocols.HttpSimpleClientProtocol class.
15	AllowAutoRedirect
16	ClientCertificates
17	ConnectionGroupName
18	Container
19	CookieContainer
20	Credentials
21	DesignMode
22	Events
23	PreAuthenticate
24	Proxy
25	RequestEncoding

r.	
diad diad diad diad diad	
Jan Jan	

1	Site
2	Timeout
3	Url
4	UserAgent
5	BeginInvoke
6	
7	[C#] protected IAsyncResult BeginInvoke(string methodName, string requestUrl,
8	object[] parameters, AsyncCallback callback, object asyncState);
9	[C++] protected: IAsyncResult* BeginInvoke(String* methodName, String*
10	requestUrl, Object* parametersgc[], AsyncCallback* callback, Object*
11	asyncState);
12	[VB] Protected Function BeginInvoke(ByVal methodName As String, ByVal
13	requestUrl As String, ByVal parameters() As Object, ByVal callback As
14	AsyncCallback, ByVal asyncState As Object) As IAsyncResult
15	[JScript] protected function BeginInvoke(methodName : String, requestUrl :
16	String, parameters: Object[], callback: AsyncCallback, asyncState: Object):
17	IAsyncResult;
18	
19	Description
20	Starts an asynchronous invocation of a method of a HTTP Web service.
21	Return Value: An System.IAsyncResult which can be passed to
22	System.Web.Services.Protocols.HttpSimpleClientProtocol.EndInvoke(System
23	.IAsyncResult) to obtain the return values from the Web Service method.
24	The methodName parameter is used to find the types of the parameters and
25	return values of the method that is invoking

System.Web.Services.Protocols.HttpSimpleClientProtocol.BeginInvoke(Syste m.String,System.String,System.Object[],System.AsyncCallback,System.Objec t) . It is also used to find custom attributes which may have been added to the method. The name of the Web Service method. The url to use when creating theSystem.Net.WebRequest. An array of objects containing the parameters to pass to the Web Service method. The order of the values in the array correspond to the order of the parameters in the calling method of the derived class. The delegate to call when the asynchronous method call is complete. If callback is null, the

EndInvoke

[C#] protected object EndInvoke(IAsyncResult asyncResult);

[C++] protected: Object\* EndInvoke(IAsyncResult\* asyncResult);

delegate is not called. Additional information supplied by a client.

[VB] Protected Function EndInvoke(ByVal asyncResult As IAsyncResult) As Object

[JScript] protected function EndInvoke(asyncResult : IAsyncResult) : Object;

## Description

Completes asynchronous invocation of a Web Service method using HTTP.

Return Value: An array of objects containing the return value and any by reference or out parameters for the Web Service method. The System.IAsyncResult returned from

System.Web.Services.Protocols.HttpSimpleClientProtocol.BeginInvoke(System.String,System.String,System.Object[],System.AsyncCallback,System.Object().

Invoke

2

3

5

6

8

9

10

12 13

> 15 16

> > 17

18 19

20

21 22

24

25

23

[C#] protected object Invoke(string methodName, string requestUrl, object[] parameters);

[C++] protected: Object\* Invoke(String\* methodName, String\* requestUrl, Object\* parameters \_ gc[]);

[VB] Protected Function Invoke(ByVal methodName As String, ByVal requestUrl As String, ByVal parameters() As Object) As Object

[JScript] protected function Invoke(methodName : String, requestUrl : String, parameters : Object[]) : Object;

#### Description

Invokes a Web Service method using HTTP.

Return Value: An array of objects containing the return value and any byreference or out parameters of the derived class method.

The *methodName* is used to find the types of the parameters and return values of the method that is invoking

System.Web.Services.Protocols.HttpSimpleClientProtocol.Invoke(System.String,System.String,System.Object[]) . It is also used to find custom attributes which may have been added to the method. The name of the Web Service method in the derived class that is invoking

System.Web.Services.Protocols.HttpSimpleClientProtocol.Invoke(System.String,System.String,System.Object[]). The URL of the Web Service method the client is requesting. An array of objects containing the parameters to pass to the

remote Web service. The order of the values in the array correspond to the order of 1 the parameters in the calling method of the derived class. 2 HttpWebClientProtocol class (System.Web.Services.Protocols) 3 **ToString** 6 Description 7 When overridden in a derived class, provides support for client proxies 8 invoking Web Services using HTTP. 9 ASP.NET incorporates two distinct functionalities of Web Services: 10 building ASP.NET Web Services and building Web Service clients. If you build a 11 Web Service client using ASP.NET, you must create a proxy class deriving 12 indirectly or directly from 13 System.Web.Services.Protocols.HttpWebClientProtocol for the Web Service 14 you want to call. 15 HttpWebClientProtocol 16 Example Syntax: 17 **ToString** 18 19 [C#] protected HttpWebClientProtocol(); 20 [C++] protected: HttpWebClientProtocol(); 21 [VB] Protected Sub New() 22 [JScript] protected function HttpWebClientProtocol(); 23 24 Description

1	Initializes a new instance of the
2	System.Web.Services.Protocols.HttpWebClientProtocol class.
3	AllowAutoRedirect
4	ToString
5	
6	[C#] public bool AllowAutoRedirect {get; set;}
7	[C++] public:property bool get_AllowAutoRedirect();public:property void
8	set_AllowAutoRedirect(bool);
9	[VB] Public Property AllowAutoRedirect As Boolean
10	[JScript] public function get AllowAutoRedirect(): Boolean; public function set
11	AllowAutoRedirect(Boolean);
12	
13	Description
14	Gets or sets whether the client automatically follows server redirects.
15	If a client sends authentification information, such as a user name and
16	password, you do not want to enable the server to redirect, because this can
17	compromise security.
18	ClientCertificates
19	ToString
20	
21	[C#] public X509CertificateCollection ClientCertificates {get;}
22	[C++] public:property X509CertificateCollection* get_ClientCertificates();
23	[VB] Public ReadOnly Property ClientCertificates As X509CertificateCollection
24	[JScript] public function get ClientCertificates(): X509CertificateCollection;
25	
	2 3 4 4 5 6 7 7 8 8 9 10 11 12 13 14 15 16 17 18 19 20 1 22 23 24 1

# Description

3

4

5

8

10

11

12

13

14

15

16

17

18

20

21

22

23

24

25

Gets the collection of client certificates.

Allows a client to pass one or more client certificates, also known as Authenticode X.509 v.3 certificates, when calling a Web Service method. If the Web Service method has been configured to use client certificates, a client certificate can be used as one mechanism for authenticating a client. For details on setting up client certificates, see the Internet Information Server (IIS) documentation.

ConnectionGroupName

Container

CookieContainer

**ToString** 

### Description

Gets or sets the collection of cookies.

If a Web Service method uses session state, then a cookie is passed back to the Web Service client that uniquely identifies the session for that Web Service client. In order for the Web Service client to receive that cookie, a new instance of System.Net.CookieContainer must be created and assigned to the System.Web.Services.Protocols.HttpWebClientProtocol.CookieContainer property before calling the Web Service method.

Credentials

DesignMode

24

25

**Events** 

PreAuthenticate

Proxy

**ToString** 

Description

Gets or sets proxy information for making a Web Service request through a firewall.

Use the **System.Web.Services.Protocols.HttpWebClientProtocol.Proxy** property if a client needs to use different proxy settings than those in the system settings. You can use the **System.Net.WebProxy** class to set the proxy settings, because it implements **System.Net.IWebProxy**.

RequestEncoding

Site

Timeout

Url

UserAgent

**ToString** 

Description

Gets or sets the value for the user agent header that is sent with each request.

The user agent string allows a Web server to identify the client.

GetWebRequest
---------------

3

4

5

7

8

ç

10

11

12

14 15

17

16

18

20

22

2324

25

[C#] protected override WebRequest GetWebRequest(Uri uri);

[C++] protected: WebRequest\* GetWebRequest(Uri\* uri);

[VB] Overrides Protected Function GetWebRequest(ByVal uri As Uri) As

WebRequest

[JScript] protected override function GetWebRequest(uri : Uri) : WebRequest;

Description

Creates a **System.Net.WebRequest** instance for the specified URI.

Return Value: The **System.Net.WebRequest** instance. The **System.Uri** for creating the **System.Net.WebRequest**.

GetWebResponse

[C#] protected override WebResponse GetWebResponse(WebRequest request);

[C++] protected: WebResponse\* GetWebResponse(WebRequest\* request);

[VB] Overrides Protected Function GetWebResponse(ByVal request As

WebRequest) As WebResponse

[JScript] protected override function GetWebResponse(request : WebRequest) :

WebResponse; Returns a response from a request to a Web Service method.

Description

Returns a response from a synchronous request to a Web Service method.

MS1-863US.APF

Return Value: The System.Net.WebResponse instance. The

System.Net.WebRequest to get the response from.

11

12

13

14

15

16

17

18

19

20

21

22

23

25

GetW	ebRes	ponse
000,	CULCU	POLIDO

[C#] protected override WebResponse GetWebResponse(WebRequest request,

IAsyncResult result);

2

[C++] protected: WebResponse\* GetWebResponse(WebRequest\* request,

IAsyncResult\* result);

[VB] Overrides Protected Function GetWebResponse(ByVal request As

WebRequest, ByVal result As IAsyncResult) As WebResponse

[JScript] protected override function GetWebResponse(request: WebRequest,

result: IAsyncResult): WebResponse;

### Description

Return Value: The System.Net.WebResponse instance. The

System.Net.WebRequest to get the response from. The System.IAsyncResult to
pass to System.Net.HttpWebRequest.EndGetResponse(System.IAsyncResult).

when the response has completed.

LogicalMethodInfo class (System.Web.Services.Protocols)

**ToString** 

#### Description

Represents the attributes and metadata for a Web Service method. This class cannot be inherited.

System.Web.Services.Protocols.LogicalMethodInfo is used primarily by a SOAP extension to interrogate the details of the Web Service method an SOAP extension is configured to run with. Depending on how the SOAP extension is configured to run, the SOAP extension can find out details about the Web Service method in the

System.Web.Services.Protocols.SoapExtension.GetInitializer(System.Web.Services.Protocols.SoapExtensionAttribute) method of System.Web.Services.Protocols.SoapExtension that takes an System.Web.Services.Protocols.LogicalMethodInfo. The System.Web.Services.Protocols.LogicalMethodInfo provides details such as the Web Service method's parameters by accessing the System.Web.Services.Protocols.LogicalMethodInfo.Parameters property and any custom attributes applied to the Web Service method using the System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetCustomAttributes(System.Web.Services.Protocols.LogicalMethodInfo.GetC

tem.Type) property.

LogicalMethodInfo

Example Syntax:

**ToString** 

[C#] public LogicalMethodInfo(MethodInfo methodInfo);

[C++] public: LogicalMethodInfo(MethodInfo\* methodInfo);

[VB] Public Sub New(ByVal methodInfo As MethodInfo)

[JScript] public function LogicalMethodInfo(methodInfo: MethodInfo); Initializes a new instance of the System.Web.Services.Protocols.LogicalMethodInfo class.

lee@hayes pik 509+324-9256 815 MS1-863US.APP

5

8

9

10

11 12 13

15 16

18

17

21 22

20

23 24

25

Description

Initializes a new instance of the

System. Web. Services. Protocols. Logical Method Info class with the System.Reflection.MethodInfo object passed. A System.Reflection.MethodInfo to initialize the properties of

System.Web.Services.Protocols.LogicalMethodInfo common to System.Reflection.MethodInfo.

> AsyncCallbackParameter **ToString**

[C#] public ParameterInfo AsyncCallbackParameter {get;} [C++] public: property ParameterInfo\* get AsyncCallbackParameter(); [VB] Public ReadOnly Property AsyncCallbackParameter As ParameterInfo [JScript] public function get AsyncCallbackParameter(): ParameterInfo;

## Description

Gets the parameter information for the AsyncCallback parameter of a Begin method in an asynchronous invocation.

The asynchronous design pattern in the common language runtime involves calling a Begin method to start the asynchronous method invocation and an End method to complete the invocation. The Begin method takes two additional parameters besides the parameters defined by the method: one for a delegate and one for any state information that needs to be passed on to the delegate. This

property represents the parameter for the delegate with a parameter name of AsyncCallback.

AsyncResultParameter

**ToString** 

5

2

3

4

6

8

9

10

12

13 14 15

16 17

> 18 19

21

20

22

24

[C#] public ParameterInfo AsyncResultParameter {get;}

[C++] public: \_\_property ParameterInfo\* get\_AsyncResultParameter();

[VB] Public ReadOnly Property AsyncResultParameter As ParameterInfo [JScript] public function get AsyncResultParameter(): ParameterInfo;

Description

Gets the return value of a Begin asynchronous method invocation.

The asynchronous design pattern in the common language runtime involves calling a **Begin** method to start the asynchronous method invocation and an **End** method to complete the invocation. The **Begin** method typically returns immediately with an object implementing the **System.IAsyncResult** interface, which can then be passed to the **End** method at a later time to complete the asynchronous method invocation. The returned object implementing the **System.IAsyncResult** interface is represented by this property.

AsyncStateParameter

**ToString** 

[C#] public ParameterInfo AsyncStateParameter {get;}

[C++] public: \_\_property ParameterInfo\* get AsyncStateParameter();

 $[VB]\ Public\ ReadOnly\ Property\ AsyncStateParameter\ As\ ParameterInfo$ 

[JScript] public function get AsyncStateParameter() : ParameterInfo;

Description

2

4

5

6

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets the parameter information for the *AsyncState* parameter of a **Begin** method in an asynchronous invocation.

The asynchronous design pattern in the common language runtime involves calling a **Begin** method to start the asynchronous method invocation and an **End** method to complete the invocation. The **Begin** method takes two additional parameters besides the parameters defined by the method: one for a delegate and one for any state information that needs to be passed on to the delegate. This property represents the state information that needs to be passed into the delegate.

BeginMethodInfo

**ToString** 

[C#] public MethodInfo BeginMethodInfo {get;}

[C++] public: \_\_property MethodInfo\* get\_BeginMethodInfo();

[VB] Public ReadOnly Property BeginMethodInfo As MethodInfo

[JScript] public function get BeginMethodInfo(): MethodInfo;

Description

Gets the attributes and metadata for a **Begin** method of an asynchronous invocation to a method.

CustomAttributeProvider

**ToString** 

1	1
1	
2	[C#] public ICustomAttributeProvider CustomAttributeProvider {get;}
3	[C++] public:property ICustomAttributeProvider*
4	get_CustomAttributeProvider();
5	[VB] Public ReadOnly Property CustomAttributeProvider As
6	ICustomAttributeProvider
7	[JScript] public function get CustomAttributeProvider():
8	ICustomAttributeProvider;
9	
10	Description
11	Gets the custom attributes applied to the method.
12	DeclaringType
13	ToString
14	
15	[C#] public Type DeclaringType {get;}
16	[C++] public:property Type* get_DeclaringType();
17	[VB] Public ReadOnly Property DeclaringType As Type
18	[JScript] public function get DeclaringType(): Type;
19	
20	Description
21	Gets the class that declares the method represented by the instance of
22	System.Web.Services.Protocols.LogicalMethodInfo.
23	The System.Web.Services.Protocols.LogicalMethodInfo.DeclaringType
24	property retrieves a reference to a System. Type for the type that declares this

member. A member of a class (or interface) is either declared or inherited from a

819

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

25

base class (or interface). The returned System. Type might not be the same as the System. Type of the class implementing the Web Service (if that class derives from a base class and the method represented by this class is declared in that base class then the **System.Type** returned is the base class). EndMethodInfo **ToString** [C#] public MethodInfo EndMethodInfo {get;} [C++] public: property MethodInfo\* get EndMethodInfo(); [VB] Public ReadOnly Property EndMethodInfo As MethodInfo [JScript] public function get EndMethodInfo(): MethodInfo; Description Gets the attributes and metadata for an End method of an asynchronous invocation to a method. InParameters **ToString** [C#] public ParameterInfo[] InParameters {get;} [C++] public: property ParameterInfo\* get InParameters(); [VB] Public ReadOnly Property InParameters As ParameterInfo () [JScript] public function get InParameters(): ParameterInfo[]; Description

1	Gets the parameters passed into the method represented by the instance of
2	System.Web.Services.Protocols.LogicalMethodInfo .
3	Use an instance of System.Reflection.ParameterInfo to obtain
4	information about the parameter's data type, default value, and so on.
5	IsAsync
6	ToString
7	
8	[C#] public bool IsAsync {get;}
9	[C++] public:property bool get_IsAsync();
10	[VB] Public ReadOnly Property IsAsync As Boolean
11	[JScript] public function get IsAsync(): Boolean;
12	
13	Description
14	Gets a value indicating whether the method represented by the instance of
15	System.Web.Services.Protocols.LogicalMethodInfo is invoked asynchronously
16	IsVoid
17	ToString
18	
19	[C#] public bool IsVoid {get;}
20	[C++] public:property bool get_IsVoid();
21	[VB] Public ReadOnly Property IsVoid As Boolean
22	[JScript] public function get IsVoid(): Boolean;
23	
24	Description
25	

	1	Gets a value indicating whether the return type for the method represented
	2	by the instance of System. Web. Services. Protocols. Logical Method Info is void.
	3	MethodInfo
	4	ToString
	5	
	6	[C#] public MethodInfo MethodInfo {get;}
	7	[C++] public:property MethodInfo* get_MethodInfo();
	8	[VB] Public ReadOnly Property MethodInfo As MethodInfo
	9	[JScript] public function get MethodInfo() : MethodInfo;
	10	
dans dans dad dam das dan dan dan dam	11	Description
And Sum	12	Gets the attributes and metadata for a synchronous method.
And Talk	13	Name
	14	ToString
	15	
	16	[C#] public string Name {get;}
	17	[C++] public:property String* get_Name();
	18	[VB] Public ReadOnly Property Name As String
	19	[JScript] public function get Name(): String;
	20	
	21	Description
	22	Gets the name of the method represented by this instance.
	23	OutParameters
	24	ToString
	25	

1	
2	[C#] public ParameterInfo[] OutParameters {get;}
3	[C++] public:property ParameterInfo* get_OutParameters();
4	[VB] Public ReadOnly Property OutParameters As ParameterInfo ()
5	[JScript] public function get OutParameters(): ParameterInfo[];
6	
7	Description
8	Gets the out parameters for the method.
9	Use an instance of System.Reflection.ParameterInfo to obtain
10	information about the parameter's data type, default value, and so on.
11	Parameters
12	ToString
13	
14	[C#] public ParameterInfo[] Parameters {get;}
15	[C++] public:property ParameterInfo* get_Parameters();
16	[VB] Public ReadOnly Property Parameters As ParameterInfo ()
17	[JScript] public function get Parameters(): ParameterInfo[];
18	
19	Description
20	Gets the parameters for the method.
21	Use an instance of System.Reflection.ParameterInfo to obtain
22	information about the parameter's data type, default value, and so on.
23	ReturnType
24	ToString
25	

1	
2	[C#] public Type ReturnType {get;}
3	[C++] public:property Type* get_ReturnType();
4	[VB] Public ReadOnly Property ReturnType As Type
5	[JScript] public function get ReturnType(): Type;
6	
7	Description
8	Gets the return type of this method.
9	ReturnTypeCustomAttributeProvider
10	ToString
11	
12	[C#] public ICustomAttributeProvider ReturnTypeCustomAttributeProvider
13	{get;}
14	[C++] public:property ICustomAttributeProvider*
15	get_ReturnTypeCustomAttributeProvider();
16	[VB] Public ReadOnly Property ReturnTypeCustomAttributeProvider As
17	ICustomAttributeProvider
18	[JScript] public function get ReturnTypeCustomAttributeProvider():
19	ICustomAttributeProvider;
20	
21	Description
22	Gets the custom attributes for the return type.
23	BeginInvoke
24	
25	[C#] public IAsyncResult BeginInvoke(object target, object[] values,

	AsyncCallback callback, object asyncState);
	[C++] public: IAsyncResult* BeginInvoke(Object* target, Object* valuesgc[],
	AsyncCallback* callback, Object* asyncState);
	[VB] Public Function BeginInvoke(ByVal target As Object, ByVal values() As
	Object, ByVal callback As AsyncCallback, ByVal asyncState As Object) As
	IAsyncResult
	[JScript] public function BeginInvoke(target : Object, values : Object[], callback :
	AsyncCallback, asyncState: Object): IAsyncResult;
	Description
	Begins an asynchronous invocation of the method represented by this
	System. Web. Services. Protocols. Logical Method Info instance.
	Return Value: An System.IAsyncResult which is passed to
	System. We b. Services. Protocols. Logical Method Info. End Invoke (System. Objective System) and the state of the state
	t, System. I Async Result) to obtain the return values from the remote method call.
	The instance of the object to invoke the method on. An argument list for the
	invoked method. This is an array of objects with the same number, order, and type

The instance of the object to invoke the method on. An argument list for the invoked method. This is an array of objects with the same number, order, and type as the parameters of the method. If the method does not require any parameters, values should be **null**. The delegate to call when the asynchronous invoke is complete. If callback is **null**, the delegate is not called. State information that is passed on to the delegate.

Create

[C#] public static LogicalMethodInfo[] Create(MethodInfo[] methodInfos); [C++] public: static LogicalMethodInfo\* Create(MethodInfo\* methodInfos[]) [];

[VB] Public Shared Function Create(ByVal methodInfos() As MethodInfo) As LogicalMethodInfo() [JScript] public static function Create(methodInfos: MethodInfo[]): LogicalMethodInfo[]; Creates an array of System. Web. Services. Protocols. Logical Method Info, given an array of System.Reflection.MethodInfo. 7 Description Creates an array of System. Web. Services. Protocols. Logical Method Info, 9 given an array of System.Reflection.MethodInfo that can contain information 10 about both asynchronous and synchronous methods. 11 Return Value: An array of System. Web. Services. Protocols. Logical Method Info, representing the methods within methodInfos. An array of 13 System.Reflection.MethodInfo representing the asynchronous and synchronous 14 methods to create System. Web. Services. Protocols. Logical Method Info objects 15 for. 16 Create 17 18 [C#] public static LogicalMethodInfo[] Create(MethodInfo[] methodInfos, 19 LogicalMethodTypes types); 20 [C++] public: static LogicalMethodInfo\* Create(MethodInfo\* methodInfos[], 21 LogicalMethodTypes types) []; 22 [VB] Public Shared Function Create(ByVal methodInfos() As MethodInfo, ByVal 23 types As LogicalMethodTypes) As LogicalMethodInfo() 24 [JScript] public static function Create(methodInfos: MethodInfo[], types:

LogicalMethodTypes) : LogicalMethodInfo[];

Description

2

4

7

8

10

11

12

13

14

15

16

17

18

19

20

21

23

Creates an array of System.Web.Services.Protocols.LogicalMethodInfo, given an array of System.Reflection.MethodInfo, where the returned array of System.Web.Services.Protocols.LogicalMethodInfo can be restricted to only asynchronous or synchronous methods.

Return Value: An array of System.Web.Services.Protocols.LogicalMethodInfo representing the methods within methodInfos, filtered by the value of types. An array of System.Reflection.MethodInfo representing the asynchronous and synchronous methods to create

System.Web.Services.Protocols.LogicalMethodInfo objects for. A bitwise combination of the System.Web.Services.Protocols.LogicalMethodTypes values. Determines whether just asynchronous or synchronous methods or both are included in the returned array of

System. Web. Services. Protocols. Logical Method Info.

EndInvoke

[C#] public object[] EndInvoke(object target, IAsyncResult asyncResult);

[C++] public: Object\* EndInvoke(Object\* target, IAsyncResult\* asyncResult)

\_\_gc[];

[VB] Public Function EndInvoke(ByVal target As Object, ByVal asyncResult As

IAsyncResult) As Object()

[JScript] public function EndInvoke(target : Object, asyncResult : IAsyncResult) :

∫∫ Object[];

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Ends an asynchronous invocation of the method represented by this System. Web. Services. Protocols. Logical Method Info instance.

Return Value: An array of objects containing the return value and any byreference or out parameters of the derived class method. The instance of the object
to invoke the method on. The System.IAsyncResult returned from

System.Web.Services.Protocols.LogicalMethodInfo.BeginInvoke(System.Object,System.Object[],System.AsyncCallback,System.Object).

**GetCustomAttribute** 

[C#] public object GetCustomAttribute(Type type);

[C++] public: Object\* GetCustomAttribute(Type\* type);

[VB] Public Function GetCustomAttribute(ByVal type As Type) As Object

[JScript] public function GetCustomAttribute(type : Type) : Object;

Description

Returns the first custom attribute applied to the type, if there are custom attributes applied to the type.

Return Value: An System.Object containing the first custom attribute applied to type. The System.Type that the custom attributes are applied to.

**GetCustomAttributes** 

[C#] public object[] GetCustomAttributes(Type type);

[C++] public: Object\* GetCustomAttributes(Type\* type) \_gc[];

	1	[VB] Public Function GetCustomAttributes(ByVal type As Type) As Object()
	2	[JScript] public function GetCustomAttributes(type : Type) : Object[];
	3	
	4	Description
	5	Returns the custom attributes applied to the specified type.
	6	Return Value: An array of System.Object containing the custom attributes ap
	7	to type. The System.Type to get the custom attributes applied to.
	8	Invoke
	9	
	10	[C#] public object[] Invoke(object target, object[] values);
	11	[C++] public: Object* Invoke(Object* target, Object* valuesgc[])gc[];
is id id	12	[VB] Public Function Invoke(ByVal target As Object, ByVal values() As Obj
	13	As Object()
	14	[JScript] public function Invoke(target : Object, values : Object[]) : Object[];
	15	
	16	Description
: d	17	Invokes the method represented by this
	18	System. Web. Services. Protocols. Logical Method Info instance.
	19	Return Value: An array of type System.Object representing the return value
	20	out parameters of the invoked method. The instance of the object to invoke the
	21	method on. An argument list for the invoked method. This is an array of obje
	22	with the same number, order, and type as the parameters of the method. If the
	23	method does not require any parameters, values should be null.
	24	IsBeginMethod
	25	

the specified type. aining the custom attributes applied ributes applied to. t[] values); Object\* values \_\_gc[]) \_\_gc[]; Object, ByVal values() As Object) , values : Object[]) : Object[]; odInfo instance. representing the return value and tance of the object to invoke the ethod. This is an array of objects arameters of the method. If the s should be **null**.

[C#] public static bool IsBeginMethod(MethodInfo methodInfo); [C++] public: static bool IsBeginMethod(MethodInfo\* methodInfo); 3 [VB] Public Shared Function IsBeginMethod(ByVal methodInfo As MethodInfo) 4 As Boolean 5 [JScript] public static function IsBeginMethod(methodInfo : MethodInfo) : Boolean; 7 8 Description Returns a value indicating whether the method passed in represents a begin 10 method of an asynchronous invocation. Return Value: true if methodInfo is a begin method of an asynchronous 12 invocation; otherwise, false . The System.Reflection.MethodInfo to determine if 13 it is a begin method of an asynchronous invocation. 14 **IsEndMethod** 15 16 [C#] public static bool IsEndMethod(MethodInfo methodInfo); 17

[C++] public: static bool IsEndMethod(MethodInfo\* methodInfo);

[VB] Public Shared Function IsEndMethod(ByVal methodInfo As MethodInfo)

As Boolean

19

20

21

22

23

24

25

[JScript] public static function IsEndMethod(methodInfo : MethodInfo) : Boolean;

Description

Returns a value indicating whether the method passed in represents a end method of an asynchronous invocation.

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Return Value: true if methodInfo is a end method of an asynchronous invocation; otherwise, false. The System.Reflection.MethodInfo to determine if it is a end method of an asynchronous invocation. **ToString** [C#] public override string ToString(); [C++] public: String\* ToString(); [VB] Overrides Public Function ToString() As String [JScript] public override function ToString() : String; Description Returns a System. String that represents the current System. Web. Services. Protocols. Logical Method Info. Return Value: A System.String that represents the current System. Web. Services. Protocols. Logical Method Info. LogicalMethodTypes enumeration (System.Web.Services.Protocols) **ToString** Description Specifies how the Web Service method was invoked. **ToString** [C#] public const LogicalMethodTypes Async;

[C++] public: const LogicalMethodTypes Async;

21 22

19

20

23 24

25

[VB] Public Const Async As LogicalMethodTypes [JScript] public var Async : LogicalMethodTypes;

Description

Specifies the Web Service method is invoked asynchronously.

**ToString** 

[C#] public const LogicalMethodTypes Sync;

[C++] public: const LogicalMethodTypes Sync;

[VB] Public Const Sync As LogicalMethodTypes

[JScript] public var Sync : LogicalMethodTypes;

Description

Specifies the Web Service method is invoked synchronously.

MatchAttribute class (System.Web.Services.Protocols)

**ToString** 

Description

Represents the attributes of a match made using text pattern matching. This class cannot be inherited.

Text pattern matching allows a Web Service to leverage existing HTML content by parsing it using regular expressions. A Web Service specifies the content it wants to parse in a Service Description using match elements. These match elements specify several items: the regular expression for parsing the

contents of an existing HTML page, whether the parsing must case-insensitive, 1 and how many instances of content that matches the regular expression should be 2 returned. When a client builds a proxy class using the Wsdl.exe tool, methods on 3 the proxy class include a System. Web. Services. Protocols. Match Attribute detailing the match elements found in the Service Description. MatchAttribute 6 Example Syntax: 7 **ToString** 8 9 [C#] public MatchAttribute(string pattern); 10 [C++] public: MatchAttribute(String\* pattern); 11 [VB] Public Sub New(ByVal pattern As String) 12 [JScript] public function MatchAttribute(pattern : String); 13 14 Description 15 Initializes a new instance of the 16 System. Web. Services. Protocols. Match Attribute class with the specified pattern. 17 Use this constructor to create and initialize a new instance of the 18 System.Web.Services.Protocols.MatchAttribute class using the specified 19 pattern. A string that represents the pattern to match. 20 Capture 21 **ToString** 22 23 [C#] public int Capture {get; set;} 24

[C++] public: \_\_property int get\_Capture();public: \_\_property void

```
set Capture(int);
1
   [VB] Public Property Capture As Integer
2
   [JScript] public function get Capture(): int;public function set Capture(int);
3
4
   Description
5
           Gets or sets a value that represents the index of a match within a grouping.
6
           Group
7
           ToString
8
9
    [C#] public int Group {get; set;}
10
    [C++] public: __property int get_Group();public: __property void set_Group(int);
11
    [VB] Public Property Group As Integer
12
    [JScript] public function get Group(): int;public function set Group(int);
13
14
    Description
15
           Gets or sets a value that represents a grouping of related matches.
16
           Use the System. Web. Services. Protocols. Match Attribute. Group property
17
    to specify a group that a match belongs to.
18
            IgnoreCase
19
            ToString
20
21
     [C#] public bool IgnoreCase {get; set;}
22
     [C++] public: __property bool get_IgnoreCase();public: __property void
23
     set_IgnoreCase(bool);
24
     [VB] Public Property IgnoreCase As Boolean
```

1	[JScript] public function get IgnoreCase(): Boolean; public function set
2	IgnoreCase(Boolean);
3	
4	Description
5	Gets or sets a value that indicates whether the pattern to match is case
6	insensitive.
7	MaxRepeats
8	ToString
9	
10	[C#] public int MaxRepeats {get; set;}
11	[C++] public:property int get_MaxRepeats();public:property void
12	set_MaxRepeats(int);
13	[VB] Public Property MaxRepeats As Integer
14	[JScript] public function get MaxRepeats(): int;public function set
15	MaxRepeats(int);
16	
17	Description
18	Gets or sets the maximum number of values to return from the match.
19	A value of 1 returns only the first match. A value of -1 returns all matches.
20	Additionally, a value of -1 equates to an '*' in a regular expression.
21	Pattern
22	ToString
23	
24	[C#] public string Pattern {get; set;}
25	[C++] public:property String* get_Pattern();public:property void

1	set_Pattern(String*);
2	[VB] Public Property Pattern As String
3	[JScript] public function get Pattern(): String; public function set Pattern(String);
4	
5	Description
6	Gets or sets a regular expression that represents the pattern to match.
7	Use the System.Web.Services.Protocols.MatchAttribute.Pattern
8	property to specify a regular expression that represents the value to match.
9	TypeId
10	MimeFormatter class (System.Web.Services.Protocols)
11	ToString
12	
13	
14	Description
15	
16	MimeFormatter
17	Example Syntax:
18	ToString
19	
20	[C#] protected MimeFormatter();
21	[C++] protected: MimeFormatter();
22	[VB] Protected Sub New()
23	[JScript] protected function MimeFormatter();
24	CreateInstance
25	

- 11	
2	[C#] public static MimeFormatter CreateInstance(Type type, object initializer);
3	[C++] public: static MimeFormatter* CreateInstance(Type* type, Object*
4	initializer);
5	[VB] Public Shared Function CreateInstance(ByVal type As Type, ByVal
6	initializer As Object) As MimeFormatter
7	[JScript] public static function CreateInstance(type : Type, initializer : Object) :
8	MimeFormatter;
9	
10	Description
11	
12	GetInitializer
13	
14	[C#] public abstract object GetInitializer(LogicalMethodInfo methodInfo);
15	[C++] public: virtual Object* GetInitializer(LogicalMethodInfo* methodInfo) = 0;
16	[VB] MustOverride Public Function GetInitializer(ByVal methodInfo As
17	LogicalMethodInfo) As Object
18	[JScript] public abstract function GetInitializer(methodInfo : LogicalMethodInfo) :
19	Object;
20	
21	Description
22	
23	GetInitializer
24	
25	[C#] public static object GetInitializer(Type type, LogicalMethodInfo

1	methodInfo);
2	[C++] public: static Object* GetInitializer(Type* type, LogicalMethodInfo*
3	methodInfo);
4	[VB] Public Shared Function GetInitializer(ByVal type As Type, ByVal
5	methodInfo As LogicalMethodInfo) As Object
6	[JScript] public static function GetInitializer(type: Type, methodInfo:
7	LogicalMethodInfo): Object;
8	
9	Description
10	
11	GetInitializers
12	
13	[C#] public virtual object[] GetInitializers(LogicalMethodInfo[] methodInfos);
14	[C++] public: virtual Object* GetInitializers(LogicalMethodInfo* methodInfos[])
15	gc[];
16	[VB] Overridable Public Function GetInitializers(ByVal methodInfos() As
17	LogicalMethodInfo) As Object()
18	[JScript] public function GetInitializers(methodInfos: LogicalMethodInfo[]):
19	Object[];
20	
21	Description
22	
23	GetInitializers
24	
25	[C#] public static object[] GetInitializers(Type type, LogicalMethodInfo[]

1	methodInfos);
2	[C++] public: static Object* GetInitializers(Type* type, LogicalMethodInfo*
3	methodInfos[])gc[];
4	[VB] Public Shared Function GetInitializers(ByVal type As Type, ByVal
5	methodInfos() As LogicalMethodInfo) As Object()
6	[JScript] public static function GetInitializers(type : Type, methodInfos :
7	LogicalMethodInfo[]): Object[];
8	
9	Description
10	
11	Initialize
12	
13	[C#] public abstract void Initialize(object initializer);
14	[C++] public: virtual void Initialize(Object* initializer) = 0;
15	[VB] MustOverride Public Sub Initialize(ByVal initializer As Object)
16	[JScript] public abstract function Initialize(initializer : Object);
17	
18	Description
19	
20	MimeParameterReader class (System.Web.Services.Protocols)
21	ToString
22	
23	
24	Description
25	

1	MimeParameterReader
2	Example Syntax:
3	ToString
4	
5	[C#] protected MimeParameterReader();
6	[C++] protected: MimeParameterReader();
7	[VB] Protected Sub New()
8	[JScript] protected function MimeParameterReader();
9	Read
10	
11	[C#] public abstract object[] Read(HttpRequest request);
12	[C++] public: virtual Object* Read(HttpRequest* request)gc[] = 0;
13	[VB] MustOverride Public Function Read(ByVal request As HttpRequest) As
14	Object()
15	[JScript] public abstract function Read(request : HttpRequest) : Object[];
16	
17	Description
18	
19	MimeParameterWriter class (System.Web.Services.Protocols)
20	ToString
21	
22	
23	Description
24	
25	MimeParameterWriter

1	Example Syntax:
2	ToString
3	
4	[C#] protected MimeParameterWriter();
5	[C++] protected: MimeParameterWriter();
6	[VB] Protected Sub New()
7	[JScript] protected function MimeParameterWriter();
8	RequestEncoding
9	ToString
10	
11	[C#] public virtual Encoding RequestEncoding {get; set;}
12	[C++] public:property virtual Encoding* get_RequestEncoding();public:
13	property virtual void set_RequestEncoding(Encoding*);
14	[VB] Overridable Public Property RequestEncoding As Encoding
15	[JScript] public function get RequestEncoding(): Encoding; public function set
16	RequestEncoding(Encoding);
17	
18	Description
19	
20	UsesWriteRequest
21	ToString
22	
23	[C#] public virtual bool UsesWriteRequest {get;}
24	[C++] public:property virtual bool get_UsesWriteRequest();
25	[VB] Overridable Public ReadOnly Property UsesWriteRequest As Boolean

1	[JScript] public function get UsesWriteRequest() : Boolean;
2	
3	Description
4	
5	GetRequestUrl
6	
7	[C#] public virtual string GetRequestUrl(string url, object[] parameters);
8	[C++] public: virtual String* GetRequestUrl(String* url, Object* parameters
9	gc[]);
10	[VB] Overridable Public Function GetRequestUrl(ByVal url As String, ByVal
11	parameters() As Object) As String
12	[JScript] public function GetRequestUrl(url : String, parameters : Object[]) :
13	String;
14	
15	Description
16	
17	InitializeRequest
18	
19	[C#] public virtual void InitializeRequest(WebRequest request, object[] values);
20	[C++] public: virtual void InitializeRequest(WebRequest* request, Object* values
21	gc[]);
22	[VB] Overridable Public Sub InitializeRequest(ByVal request As WebRequest,
23	ByVal values() As Object)
24	[JScript] public function InitializeRequest(request : WebRequest, values :
25	Object[]);

1	
2	Description
3	
4	WriteRequest
5	
6	[C#] public virtual void WriteRequest(Stream requestStream, object[] values);
7	[C++] public: virtual void WriteRequest(Stream* requestStream, Object* values
8	gc[]);
9	[VB] Overridable Public Sub WriteRequest(ByVal requestStream As Stream,
10	ByVal values() As Object)
11	[JScript] public function WriteRequest(requestStream : Stream, values : Object[]);
12	
13	Description
14	
15	MimeReturnReader class (System.Web.Services.Protocols)
16	WriteRequest
17	
18	
19	Description
20	
21	MimeReturnReader
22	Example Syntax:
23	WriteRequest
24	
25	[C#] protected MimeReturnReader();

1	[C++] protected: MimeReturnReader();
2	[VB] Protected Sub New()
. 3	[JScript] protected function MimeReturnReader();
4	Read
5	
6	[C#] public abstract object Read(WebResponse response, Stream
7	responseStream);
8	[C++] public: virtual Object* Read(WebResponse* response, Stream*
9	responseStream) = 0;
10	[VB] MustOverride Public Function Read(ByVal response As WebResponse,
11	ByVal responseStream As Stream) As Object
12	[JScript] public abstract function Read(response : WebResponse, responseStream :
13	Stream): Object;
14	
15	Description
16	
17	NopReturnReader class (System.Web.Services.Protocols)
18	ToString
19	
20	
21	Description
22	
23	NopReturnReader
24	Example Syntax:
25	ToString

1	
2	[C#] public NopReturnReader();
3	[C++] public: NopReturnReader();
4	[VB] Public Sub New()
5	[JScript] public function NopReturnReader();
6	GetInitializer
7	
8	[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);
9	[C++] public: Object* GetInitializer(LogicalMethodInfo* methodInfo);
10	[VB] Overrides Public Function GetInitializer(ByVal methodInfo As
11	LogicalMethodInfo) As Object
12	[JScript] public override function GetInitializer(methodInfo: LogicalMethodInfo)
13	: Object;
14	
15	Description
16	
17	Initialize
18	
19	[C#] public override void Initialize(object initializer);
20	[C++] public: void Initialize(Object* initializer);
21	[VB] Overrides Public Sub Initialize(ByVal initializer As Object)
22	[JScript] public override function Initialize(initializer : Object);
23	
24	Description
25	

1	Read
2	
3	[C#] public override object Read(WebResponse response, Stream
4	responseStream);
5	[C++] public: Object* Read(WebResponse* response, Stream* responseStream);
6	[VB] Overrides Public Function Read(ByVal response As WebResponse, ByVal
7	responseStream As Stream) As Object
8	[JScript] public override function Read(response : WebResponse, responseStream
9	: Stream) : Object;
10	
11	Description
12	
13	PatternMatcher class (System.Web.Services.Protocols)
14	ToString
15	
16	
17	Description
18	
19	PatternMatcher
20	Example Syntax:
21	ToString
22	
23	[C#] public PatternMatcher(Type type);
24	[C++] public: PatternMatcher(Type* type);
25	[VB] Public Sub New(ByVal type As Type)

1	[JScript] public function PatternMatcher(type: Type);
2	
3	Description
4	
5	Match
6	
7	[C#] public object Match(string text);
8	[C++] public: Object* Match(String* text);
9	[VB] Public Function Match(ByVal text As String) As Object
10	[JScript] public function Match(text : String) : Object;
11	
12	Description
13	
14	SoapClientMessage class (System.Web.Services.Protocols)
15	ToString
16	
17	
18	Description
19	Represents the data in a SOAP request sent or a SOAP response received
20	by a Web Service client at a specific
21	System.Web.Services.Protocols.SoapMessageStage for a Web Service method.
22	Action
23	ToString
24	
25	[C#] public override string Action {get;}

1	[C++] public:property virtual String* get_Action();
2	[VB] Overrides Public ReadOnly Property Action As String
3	[JScript] public function get Action(): String;
4	
5	Description
6	Gets the SOAPAction HTTP request header field for the SOAP request or
7	SOAP response.
8	The System.Web.Services.Protocols.SoapClientMessage.Action property
9	is availavable in any System. Web. Services. Protocols. Soap Message Stage.
10	Client
11	ToString
12	
13	[C#] public SoapHttpClientProtocol Client {get;}
14	[C++] public:property SoapHttpClientProtocol* get_Client();
15	[VB] Public ReadOnly Property Client As SoapHttpClientProtocol
16	[JScript] public function get Client() : SoapHttpClientProtocol;
17	
18	Description
19	Gets an instance of the client proxy class, which derives from
20	System.Web.Services.Protocols.SoapHttpClientProtocol.
21	The System.Web.Services.Protocols.SoapClientMessage.Client property
22	can be accessed during any System.Web.Services.Protocols.SoapMessageStage
23	•
24	ContentType
25	Exception

4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	١
17	
18	
19	
20	
21	
22	
23	

3

Headers
MethodInfo
ToString
Description
Gets a representation of the method prototype for the Web Service method
for which the SOAP request is intended.
Although the
$System. Web. Services. Protocols. Soap Client Message. Method Info \ property \ can be a support to the contraction of the property of the contraction of the contr$
be accessed during any stage, the method information is only available during
System. Web. Services. Protocols. So ap Message Stage. After Description and the support of th
$System. Web. Services. Protocols. Soap Message Stage. Before Serialize \ .$
OneWay
ToString
[C#] public override bool OneWay {get;}
[C++] public:property virtual bool get_OneWay();
[VB] Overrides Public ReadOnly Property OneWay As Boolean
[JScript] public function get OneWay(): Boolean;
Description
O

Gets a value indicating whether the client waits for the server to finish processing a Web Service method.

25

Description

A System.Web.Services.Protocols.SoapDocumentMethodAttribute or
System.Web.Services.Protocols.SoapRpcMethodAttribute applied to a Web
Service method or Web Service client can specify whether the Web Service
method is one way or not by setting the
System. We b. Services. Protocols. So ap Document Method Attribute. One Way
property of the attribute.
Stage
Stream
Url
ToString
Description
Gets the URL of the Web Service.
The System.Web.Services.Protocols.SoapClientMessage.Url property
can be accessed during any System.Web.Services.Protocols.SoapMessageStage
•
EnsureInStage
[C#] protected override void EnsureInStage();
[C++] protected: void EnsureInStage();
[VB] Overrides Protected Sub EnsureInStage()
[JScript] protected override function EnsureInStage();

2

3

7

8

9

10

11

13

14

15

16

17

18

19

20

21

22

23

24

25

Asserts that the current

**System.Web.Services.Protocols.SoapMessageStage** stage is a stage where in parameters are available. If not, an exception is thrown.

For Web Service clients, the in parameters are available prior to the SOAP request serialization process in the

System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize stage. The System.InvalidOperationException is thrown if

 ${\bf System. Web. Services. Protocols. Soap Client Message. Ensure In Stage \ method \ is}$  invoked in any other {\bf System. Web. Services. Protocols. Soap Message Stage}\ .

EnsureOutStage

[C#] protected override void EnsureOutStage();

[C++] protected: void EnsureOutStage();

[VB] Overrides Protected Sub EnsureOutStage()

[JScript] protected override function EnsureOutStage();

Description

Asserts that the current

**System.Web.Services.Protocols.SoapMessageStage** stage is a stage where out parameters are available. If not, an exception is thrown.

For Web Service clients, the out parameters are available after the SOAP response descrialization process in the

System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize stage. The System.InvalidOperationException is thrown if

System.Web.Services.Protocols.SoapClientMessage.EnsureOutStage method is invoked in any other System.Web.Services.Protocols.SoapMessageStage .

SoapDocumentMethodAttribute class (System.Web.Services.Protocols)

ToString

Description

Applying the optional

System.Web.Services.Protocols.SoapDocumentMethodAttribute to a Web

Service method alters the format of the SOAP request or response sent to and from a Web Service method.

Web Services Description Language (WSDL) defines two styles for how a

Web Service method, which it calls an operation, can be encoded in a SOAP

Web Services Description Language (WSDL) defines two styles for how a Web Service method, which it calls an operation, can be encoded in a SOAP request or a SOAP response: RPC and Document. The Document style refers to encoding the Web Service method as described in an XSD schema. If Document style is used, the Service Description for the Web Service, which is in WSDL, defines XSD schemas for both SOAP requests and SOAP responses to the Web Service method. A Web Service set to the Document encoding style expects clients to pass the XML just as it is defined in these XSD schemas.

Soap Document Method Attribute

Example Syntax:

**ToString** 

[C#] public SoapDocumentMethodAttribute();

[C++] public: SoapDocumentMethodAttribute();

1	[VB] Public Sub New()
2	[JScript] public function SoapDocumentMethodAttribute(); Initializes a new
3	instance of the
4	System.Web.Services.Protocols.SoapDocumentMethodAttribute class.
5	
6	Description
7	Initializes a new instance of the
8	System.Web.Services.Protocols.SoapDocumentMethodAttribute class.
9	SoapDocumentMethodAttribute
10	Example Syntax:
11	ToString
12	
13	[C#] public SoapDocumentMethodAttribute(string action);
14	[C++] public: SoapDocumentMethodAttribute(String* action);
15	[VB] Public Sub New(ByVal action As String)
16	[JScript] public function SoapDocumentMethodAttribute(action : String);
17	
18	Description
19	Initializes a new instance of the
20	System.Web.Services.Protocols.SoapDocumentMethodAttribute class setting
21	the System.Web.Services.Protocols.SoapDocumentMethodAttribute.Action
22	property to action. The intent of the SOAP request. Sets the
23	System.Web.Services.Protocols.SoapDocumentMethodAttribute.Action
24	property.
25	Action

1	ToString
2	
3	[C#] public string Action {get; set;}
4	[C++] public:property String* get_Action();public:property void
5	set_Action(String*);
6	[VB] Public Property Action As String
7	[JScript] public function get Action(): String; public function set Action(String);
8	
9	Description
10	Gets or sets the intent of the SOAP request.
11	The
12	System.Web.Services.Protocols.SoapDocumentMethodAttribute.Action
13	property forms the SOAPAction HTTP Header Field for the HTTP Request.
14	Binding
15	ToString
16	
17	[C#] public string Binding {get; set;}
18	[C++] public:property String* get_Binding();public:property void
19	set_Binding(String*);
20	[VB] Public Property Binding As String
21	[JScript] public function get Binding(): String; public function set Binding(String)
22	
23	Description
24	Gets or sets the binding a Web Service method is implementing a operation
25	for.

2

3

5

6

7

8

9

10

11

12

13

15

17

18

19

20

21

22

23

24

25

A binding, as defined by Web Services Description Language(WSDL), is similar to an interface, in that it defines a concrete set of operations. With respect to ASP.NET Web Services, each Web Service method is an operation within a binding. Web Service methods are members of either the default binding for a Web Service or in a binding specified within a

System.Web.Services.WebServiceBindingAttribute applied to a Web Service.

A Web Service can implement multiple bindings, by applying multiple

System.Web.Services.WebServiceBindingAttribute attributes to a Web Service.

OneWay

**ToString** 

[C#] public bool OneWay {get; set;}

[C++] public: \_\_property bool get\_OneWay();public: \_\_property void set\_OneWay(bool);

[VB] Public Property OneWay As Boolean

[JScript] public function get OneWay(): Boolean; public function set OneWay(Boolean);

Description

Gets or sets whether a Web Service client waits for the Web server to finish processing a Web Service method.

When a Web Service method sets the

System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay
property to true, the Web Service client does not have to wait for the Web server
to finish processing the Web Service method. As soon as the Web server has

deserialized the System.Web.Services.Protocols.SoapServerMessage, but before invoking the Web Service method, the server returns an HTTP 202 status code. A HTTP 202 status code indicates to the client that the Web server has started processing of the message. Therefore, a Web Service client receives no acknowledgment that the Web server successfully processed the message.

ParameterStyle

ToString

[C#] public SoapParameterStyle ParameterStyle {get; set;}

[C++] public: \_\_property SoapParameterStyle get\_ParameterStyle();public: \_\_property void set\_ParameterStyle(SoapParameterStyle);

[VB] Public Property ParameterStyle As SoapParameterStyle

[JScript] public function get ParameterStyle() : SoapParameterStyle;public function set ParameterStyle(SoapParameterStyle);

### Description

Gets or sets whether parameters are wrapped within a single element beneath the **Body** element in the XML portion of a SOAP message.

RequestElementName

**ToString** 

[C#] public string RequestElementName {get; set;}
[C++] public: \_\_property String\* get\_RequestElementName();public: \_\_property
void set\_RequestElementName(String\*);
[VB] Public Property RequestElementName As String

3

4

5

6

7

8

10

11

12

13

15

16

17

18

19

20

21

22

23

24

25

[JScript] public function get RequestElementName(): String; public function set RequestElementName(String); Description Gets or sets the XML element associated with the SOAP request for a Web Service method. The System. Web. Services. Protocols. Soap Document Method Attribute. Request Element Method Attribute. The support of the protocol of the protomentName defines the XML element used to wrap the parameters beneath the **Body** element of the SOAP request when System. Web. Services. Protocols. Soap Document Method Attribute. Parameter State of the protocol of the provle is System. Web. Services. Protocols. Soap Parameter Style. Wrapped. This is reflected in the XSD schema representing the SOAP request to the Web Service method within the Web Service's Service Description. RequestNamespace **ToString** [C#] public string RequestNamespace {get; set;} [C++] public: \_\_property String\* get\_RequestNamespace();public: \_\_property void set RequestNamespace(String\*); [VB] Public Property RequestNamespace As String [JScript] public function get RequestNamespace(): String; public function set RequestNamespace(String); Description

Gets or sets the namespace associated with the SOAP request for a Web Service method.

System.Web.Services.Protocols.SoapDocumentMethodAttribute.Reque stNamespace is used in the XSD schema for the Web Service method within its Service Description.

ResponseElementName

**ToString** 

[C#] public string ResponseElementName {get; set;}

[C++] public: \_\_property String\* get\_ResponseElementName();public: \_\_property void set\_ResponseElementName(String\*);

[VB] Public Property ResponseElementName As String

[JScript] public function get ResponseElementName(): String;public function set ResponseElementName(String);

Description

Gets or sets the XML element associated with the SOAP response for a Web Service method.

The

System.Web.Services.Protocols.SoapDocumentMethodAttribute.ResponseEle
mentName defines the XML element used to wrap the parameters beneath the
Body element of the SOAP response when

System. Web. Services. Protocols. So ap Document Method Attribute. Parameter Style is System. Web. Services. Protocols. So ap Parameter Style. Wrapped . This is

1	reflected in the XSD schema representing the SOAP response to the Web Service
2	method within the Web Service's Service Description.
3	ResponseNamespace
4	ToString
5	
6	[C#] public string ResponseNamespace {get; set;}
7	[C++] public:property String* get_ResponseNamespace();public:property
8	<pre>void set_ResponseNamespace(String*);</pre>
9	[VB] Public Property ResponseNamespace As String
10	[JScript] public function get ResponseNamespace(): String; public function set
11	ResponseNamespace(String);
12	
13	Description
14	Gets or sets the XML namespace associated with the SOAP response for a
15	Web Service method.
16	The
17	System.Web.Services.Protocols.SoapDocumentMethodAttribute.ResponseNa
18	mespace property is used in the XSD schema for the Web Service method in its
19	Service Description.
20	TypeId
21	Use
22	ToString
23	
24	
25	Description

Gets or sets the parameter encoding for a Web Service method within the XML portion of a SOAP message.

The Web Services Description Language (WSDL) defines two parameter encoding styles: System.Web.Services.Description.SoapBindingUse.Encoded and System.Web.Services.Description.SoapBindingUse.Literal.

SoapDocumentServiceAttribute class (System.Web.Services.Protocols)
ToString

#### Description

Applying the optional

System.Web.Services.Protocols.SoapDocumentServiceAttribute to a Web Service sets the default format of SOAP requests and responses sent to and from Web Service methods within the Web Service.

The System.Web.Services.Protocols.SoapDocumentServiceAttribute allows you to set the default encoding styles for Web Service methods within a Web Service. If an individual Web Service method needs to change these defaults, apply a System.Web.Services.Protocols.SoapDocumentMethodAttribute to that Web Service method.

SoapDocumentServiceAttribute

Example Syntax:

**ToString** 

[C#] public SoapDocumentServiceAttribute();

[C++] public: SoapDocumentServiceAttribute();

1	[VB] Public Sub New()
2	[JScript] public function SoapDocumentServiceAttribute(); Initializes a new
3	instance of the
4	System.Web.Services.Protocols.SoapDocumentServiceAttribute class.
5	
6	Description
7	Initializes a new instance of the
8	System.Web.Services.Protocols.SoapDocumentServiceAttribute class setting
9	all properties to their defaults.
10	SoapDocumentServiceAttribute
11	Example Syntax:
12	ToString
13	
14	[C#] public SoapDocumentServiceAttribute(SoapBindingUse use);
15	[C++] public: SoapDocumentServiceAttribute(SoapBindingUse use);
16	[VB] Public Sub New(ByVal use As SoapBindingUse)
17	[JScript] public function SoapDocumentServiceAttribute(use : SoapBindingUse);
18	
19	Description
20	Initializes a new instance of the
21	System.Web.Services.Protocols.SoapDocumentServiceAttribute class setting
22	the parameter encoding. The parameter encoding of the Web Service. Sets the
23	System.Web.Services.Protocols.SoapDocumentServiceAttribute.Use property.
24	SoapDocumentServiceAttribute
25	Example Syntax:

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

ToString

[C#] public SoapDocumentServiceAttribute(SoapBindingUse use,

SoapParameterStyle paramStyle);

[C++] public: SoapDocumentServiceAttribute(SoapBindingUse use,

SoapParameterStyle paramStyle);

[VB] Public Sub New(ByVal use As SoapBindingUse, ByVal paramStyle As

SoapParameterStyle)

[JScript] public function SoapDocumentServiceAttribute(use: SoapBindingUse,

paramStyle: SoapParameterStyle);

#### Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapDocumentServiceAttribute class setting the parameter encoding and whether parameters are wrapped within a single element under the **Body** element within the XML portion of SOAP requests and responses. The parameter encoding style. Sets the

 ${\bf System. Web. Services. Protocols. Soap Document Service Attribute. Use \ property.}$ 

Sets whether parameters are wrapped within a single XML element under the **Body** element within the XML portion of SOAP requests and responses to Web Service methods within the Web Service. Sets the

System.Web.Services.Protocols.SoapDocumentServiceAttribute.ParameterSt yle property.

ParameterStyle

**ToString** 

```
1
    [C#] public SoapParameterStyle ParameterStyle {get; set;}
2
    [C++] public: property SoapParameterStyle get ParameterStyle();public:
3
      property void set ParameterStyle(SoapParameterStyle);
4
    [VB] Public Property ParameterStyle As SoapParameterStyle
5
    [JScript] public function get ParameterStyle(): SoapParameterStyle; public
6
    function set ParameterStyle(SoapParameterStyle);
7
8
    Description
9
           Gets or sets the default for Web Service methods within the Web Service
10
    whether parameters are wrapped within a single element beneath the Body
11
    element in the XML portion of a SOAP message.
12
           RoutingStyle
13
           ToString
14
15
    [C#] public SoapServiceRoutingStyle RoutingStyle {get; set;}
16
    [C++] public: property SoapServiceRoutingStyle get RoutingStyle();public:
17
      property void set RoutingStyle(SoapServiceRoutingStyle);
18
    [VB] Public Property RoutingStyle As SoapServiceRoutingStyle
19
    [JScript] public function get RoutingStyle(): SoapServiceRoutingStyle; public
20
    function set RoutingStyle(SoapServiceRoutingStyle);
21
22
    Description
23
24
           TypeId
25
```

Use

**ToString** 

Description

Gets or sets the default parameter encoding for a Web Service.

The Web Services Description Language (WSDL) defines two parameter encoding styles: System.Web.Services.Description.SoapBindingUse.Encoded and System.Web.Services.Description.SoapBindingUse.Literal.

SoapException class (System.Web.Services.Protocols)
ToString

## Description

The exception that is thrown when a Web Service method is called over Simple Object Access Protocol (SOAP) and an exception occurs.

The System.Web.Services.Protocols.SoapException can either be thrown by the common language runtime or by a Web Service method. The common language runtime can throw a System.Web.Services.Protocols.SoapException if a response to a request is not formatted correctly. Web Service methods can generate a System.Web.Services.Protocols.SoapException by simply throwing an exception within the Web Service method. If the client accessed the method over SOAP, the exception is caught on the server and wrapped inside a new System.Web.Services.Protocols.SoapException . The SoapException thrown has

• •

the following property values: Property Value System. Exception. Message The **System.**Exception. Message property of the original exception.

**ToString** 

2

3

8

9

10

12

13

14

15

16

17

18

[C#] public static readonly XmlQualifiedName ClientFaultCode;

[C++] public: static XmlQualifiedName\* ClientFaultCode;

[VB] Public Shared ReadOnly ClientFaultCode As XmlQualifiedName

[JScript] public static var ClientFaultCode : XmlQualifiedName;

Description

Specifies a SOAP fault code representing a client call was not formatted correctly or did not contain the appropriate information.

An example of when a

System.Web.Services.Protocols.SoapException.ClientFaultCode can be generated, is when a client call lacks proper authentication or payment information. It is generally an indication that the client call should not be resent without change.

**ToString** 

19

20

21

22

23

24

[C#] public static readonly XmlQualifiedName DetailElementName;

[C++] public: static XmlQualifiedName\* DetailElementName;

[VB] Public Shared ReadOnly DetailElementName As XmlQualifiedName

[JScript] public static var DetailElementName : XmlQualifiedName;

Description

18

19

20

21

22

23

24

25

1

2

3

4

5

6

7

8

9

Gets an System.Xml.XmlQualifiedName representing the System.Web.Services.Protocols.SoapException.Detail element of a SOAP Fault code.

System. Web. Services. Protocols. Soap Exception. Detail property, the System.Xml.XmlQualifiedName.Name and System.Xml.XmlQualifiedName.Namespace properties of System.Web.Services.Protocols.SoapException.DetailElementName can be

In building an System.Xml.XmlNode for the

used to ensure consistancy with the SOAP specification.

**ToString** 

[C#] public static readonly XmlQualifiedName MustUnderstandFaultCode; [C++] public: static XmlQualifiedName\* MustUnderstandFaultCode; [VB] Public Shared ReadOnly MustUnderstandFaultCode As XmlQualifiedName [JScript] public static var MustUnderstandFaultCode : XmlQualifiedName;

Description

A SOAP Fault Code representing a SOAP element marked with the MustUnderstand attribute was not processed.

Not all SOAP elements require processing by the server. However, if a SOAP element is marked with the MustUnderstand attribute equal to 1, processing is required. Failure to process the element, generates a

System.Web.Services.Protocols.SoapException with a

 $System. Web. Services. Protocols. Soap Exception. Must Understand Fault Code \ .$ 

**ToString** 

23

[C#] public static readonly XmlQualifiedName ServerFaultCode;

[C++] public: static XmlQualifiedName\* ServerFaultCode;

[VB] Public Shared ReadOnly ServerFaultCode As XmlQualifiedName

[JScript] public static var ServerFaultCode : XmlQualifiedName;

### Description

Specifies a SOAP fault code representing an error occurred during the processing of a client call on the server, where the problem was not due to the message contents.

A System.Web.Services.Protocols.SoapException.ServerFaultCode might occur for example, if a server being called, couldn't respond to a request due to network problems. Typically, with this type of exception, the client call may succeed later.

**ToString** 

[C#] public static readonly XmlQualifiedName VersionMismatchFaultCode;

[C++] public: static XmlQualifiedName\* VersionMismatchFaultCode;

[VB] Public Shared ReadOnly VersionMismatchFaultCode As

XmlQualifiedName

[JScript] public static var VersionMismatchFaultCode : XmlQualifiedName;

#### Description

A SOAP fault code representing an invalid namespace for a SOAP Envelope was found during the processing of the SOAP message.

23

24

A SOAP Envelope is the top-level element of a XML document representing a SOAP message.

SoapException

Example Syntax:

**ToString** 

[C#] public SoapException(string message, XmlQualifiedName code);

[C++] public: SoapException(String\* message, XmlQualifiedName\* code);

[VB] Public Sub New(ByVal message As String, ByVal code As

XmlQualifiedName)

[JScript] public function SoapException(message : String, code :

XmlQualifiedName);

Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapException class that sets the exception message and exception code. A message that identifies the reason the exception occurred. Sets the System.Exception.Message property. The type of error that occurred. Sets the System.Web.Services.Protocols.SoapException.Code property.

SoapException

Example Syntax:

**ToString** 

[C#] public SoapException(string message, XmlQualifiedName code, Exception

22

23

21

24

25

innerException);

[C++] public: SoapException(String\* message, XmlQualifiedName\* code,

Exception\* innerException);

[VB] Public Sub New(ByVal message As String, ByVal code As

XmlQualifiedName, ByVal innerException As Exception)

[JScript] public function SoapException(message : String, code :

XmlQualifiedName, innerException: Exception);

Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapException class that sets the exception message, exception code and a reference to the root cause of the exception. A message that identifies the reason the exception occurred. Sets the System.Exception.Message property. The type of error that occurred. Sets the System.Web.Services.Protocols.SoapException.Code property. A reference to the root cause of the exception. Sets the System.Exception.InnerException property.

SoapException

Example Syntax:

**ToString** 

[C#] public SoapException(string message, XmlQualifiedName code, string actor);

[C++] public: SoapException(String\* message, XmlQualifiedName\* code, String\* actor);

21

23

24

lee@haves plic 509+324+9256

[VB] Public Sub New(ByVal message As String, ByVal code As

XmlQualifiedName, ByVal actor As String)

[JScript] public function SoapException(message : String, code :

XmlQualifiedName, actor: String); Initializes a new instance of the

System. Web. Services. Protocols. Soap Exception class.

Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapException class that sets the exception message, exception code and the piece of code that caused the exception. A message that identifies the reason the exception occurred. Sets the System.Exception.Message property. The type of error that occurred. Sets the System.Web.Services.Protocols.SoapException.Code property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the System.Web.Services.Protocols.SoapException.Actor property.

SoapException

Example Syntax:

**ToString** 

[C#] public SoapException(string message, XmlQualifiedName code, string actor,

Exception innerException);

[C++] public: SoapException(String\* message, XmlQualifiedName\* code,

String\* actor, Exception\* innerException);

[VB] Public Sub New(ByVal message As String, ByVal code As

XmlQualifiedName, ByVal actor As String, ByVal innerException As Exception)

20

21

22

23

24

1

2

[JScript] public function SoapException(message : String, code : XmlQualifiedName, actor: String, innerException: Exception); Description Initializes a new instance of the System. Web. Services. Protocols. Soap Exception class that sets the exception message, exception code, the piece of code that casued the exception and a reference to the root cause of the exception. A message that identifies the reason the exception occurred. Sets the System. Exception. Message property. The type of error that occurred. Sets the System. Web. Services. Protocols. Soap Exception. Code property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the System. Web. Services. Protocols. Soap Exception. Actor property. A reference to the root cause of an exception. Sets the System. Exception. Inner Exception property. SoapException Example Syntax: **ToString** [C#] public SoapException(string message, XmlQualifiedName code, string actor, XmlNode detail);

[C++] public: SoapException(String\* message, XmlQualifiedName\* code,

String\* actor, XmlNode\* detail); [VB] Public Sub New(ByVal message As String, ByVal code As

XmlQualifiedName, ByVal actor As String, ByVal detail As XmlNode)

20

21

22

23

24

25

[JScript] public function SoapException(message : String, code :

XmlQualifiedName, actor: String, detail: XmlNode);

Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapException class that sets the exception message, exception code, piece of code that caused the exception and application specific exception information. A message that identifies the reason the exception occurred. Sets the System.Exception.Message property. The type of error that occurred. Sets the System.Web.Services.Protocols.SoapException.Code property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the System.Web.Services.Protocols.SoapException.Actor property. The application specific exception information. Sets the System.Web.Services.Protocols.SoapException.Detail property.

SoapException

Example Syntax:

**ToString** 

[C#] public SoapException(string message, XmlQualifiedName code, string actor,

XmlNode detail, Exception innerException);

[C++] public: SoapException(String\* message, XmlQualifiedName\* code,

String\* actor, XmlNode\* detail, Exception\* innerException);

[VB] Public Sub New(ByVal message As String, ByVal code As

XmlQualifiedName, ByVal actor As String, ByVal detail As XmlNode, ByVal

innerException As Exception)

1

20 21

19

22

23

24

25

[JScript] public function SoapException(message : String, code :

XmlQualifiedName, actor: String, detail: XmlNode, innerException: Exception);

Description

Initializes a new instance of the

**System.Web.Services.Protocols.SoapException** class that sets the exception message, exception code, piece of code that caused the exception, application specific exception information and a reference to the root cause of the exception.

A message that identifies the reason the exception occurred. Sets the

System.Exception.Message property. The type of error that occurred. Sets the System.Web.Services.Protocols.SoapException.Code property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the System.Web.Services.Protocols.SoapException.Actor property. The application specific exception information. Sets the

**System.Web.Services.Protocols.SoapException.Detail** property. A reference to the root cause of the exception. Sets the **System.Exception.InnerException** property.

SoapException

Example Syntax:

**ToString** 

[C#] public SoapException(string message, XmlQualifiedName code, string actor, XmlNode detail, XmlNode[] otherElements);

[C++] public: SoapException(String\* message, XmlQualifiedName\* code, String\* actor, XmlNode\* detail, XmlNode\* otherElements[]);

	1	[V
	2	Xn
	3	oth
	4	[JS
	5	Xn
	6	Xr
	7	
	8	De
	9	
	10	Sy
	11	me
	12	spe
	13	ide
	14	pro
	15	Sy
	16	со
	17	the
	18	ap
	19	Sy
	20	ex

23

24

ToString

[VB] Public Sub New(ByVal message As String, ByVal code As
XmlQualifiedName, ByVal actor As String, ByVal detail As XmlNode, ByVal
otherElements() As XmlNode)
[JScript] public function SoapException(message : String, code :
XmlQualifiedName, actor: String, detail: XmlNode, otherElements:
XmlNode[]);
Description
Initializes a new instance of the
System.Web.Services.Protocols.SoapException class that sets the exception
message, exception code, piece of code that caused the exception, application
specific exception information and optional exception information. A message that
identifies the reason the exception occurred. Sets the System.Exception.Message
property. The type of error that occurred. Sets the
System.Web.Services.Protocols.SoapException.Code property. The piece of
code that caused the exception. Typically, a URL to a Web Service method. Sets
the System.Web.Services.Protocols.SoapException.Actor property. The
application specific exception information. Sets the
System.Web.Services.Protocols.SoapException.Detail property. Optional
exception information. Sets the
${\bf System. Web. Services. Protocols. Soap Exception. Other Elements\ property.}$
SoapException
Example Syntax:

lee@hayes pitc 509-324-9256 874 MS1-863US.APP

19

20

21

22

23

24

[C#] public SoapException(string message, XmlQualifiedName code, string actor, XmlNode detail, XmlNode[] otherElements, Exception innerException); [C++] public: SoapException(String\* message, XmlQualifiedName\* code, String\* actor, XmlNode\* detail, XmlNode\* otherElements[], Exception\* innerException); [VB] Public Sub New(ByVal message As String, ByVal code As XmlQualifiedName, ByVal actor As String, ByVal detail As XmlNode, ByVal otherElements() As XmlNode, ByVal innerException As Exception) [JScript] public function SoapException(message : String, code : XmlQualifiedName, actor: String, detail: XmlNode, otherElements: XmlNode[], innerException: Exception);

#### Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapException class that sets the exception message, exception code, piece of code that caused the exception, application specific exception information, optional exception information, and a reference to the root cause of the exception. A message that identifies the reason the exception occurred. Sets the System. Exception. Message property. The type of error that occurred. Sets the System. Web. Services. Protocols. Soap Exception. Code property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the System. Web. Services. Protocols. Soap Exception. Actor property. The application specific exception information. Sets the System. Web. Services. Protocols. Soap Exception. Detail property. Optional

```
exception information. Sets the
   1
      System. Web. Services. Protocols. Soap Exception. Other Elements property. A
   2
      reference to the root cause of the exception. Sets the
   3
      System.Exception.InnerException property.
   4
             Actor
   5
             ToString
   6
   7
       [C#] public string Actor {get;}
   8
       [C++] public: property String* get_Actor();
   9
       [VB] Public ReadOnly Property Actor As String
11 12 13
       [JScript] public function get Actor(): String;
       Description
             Gets the piece of code that caused the exception.
             System.Web.Services.Protocols.SoapException.Actor can only be set
       using one of the constructors that accept an
       System.Web.Services.Protocols.SoapException.Actor argument.
  17
             Code
  18
             ToString
  19
  20
       [C#] public XmlQualifiedName Code {get;}
  21
       [C++] public: property XmlQualifiedName* get Code();
  22
       [VB] Public ReadOnly Property Code As XmlQualifiedName
  23
       [JScript] public function get Code(): XmlQualifiedName;
  24
  25
```

21

22

23

24

25

Description

Gets the type of SOAP fault code.

**System.Web.Services.Protocols.SoapException.Code** can only be set when creating a new instance of the

System. Web. Services. Protocols. Soap Exception class.

Detail

**ToString** 

[C#] public XmlNode Detail {get;}

[C++] public: property XmlNode\* get\_Detail();

[VB] Public ReadOnly Property Detail As XmlNode

[JScript] public function get Detail(): XmlNode;

Description

Gets an **System.Xml.XmlNode** representing the application specific error information.

**System.Web.Services.Protocols.SoapException.Detail** can be set using one of the class constructors that accept a

 ${\bf System. Web. Services. Protocols. Soap Exception. Detail \ {\bf value}.}$ 

HelpLink

**HResult** 

InnerException

Message

OtherElements

	1	-
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
	10	
	11	
	12	
	13	
	14	
	15	
	16	
ļ. ak	17	
	18	
	19	
	20	
	٥.	ll .

23

24

**ToString** 

Description

Gets an array of **System.Xml.XmlNode** with the optional error information.

System.Web.Services.Protocols.SoapException.OtherElements can be set using one of the class constructors that accept an argument for System.Web.Services.Protocols.SoapException.OtherElements.

Source

StackTrace

TargetSite

SoapExtension class (System.Web.Services.Protocols)

**ToString** 

Description

Represents the ase class for SOAP extensions in ASP.NET Web Services.

ASP.NET Web Services allows a SOAP-related infrastructure to be built by means of an extensibility mechanism. The ASP.NET SOAP extension architecture revolves around an extension that can inspect or modify a message at specific stages in message processing on either the client or the server.

SoapExtension

Example Syntax:

**ToString** 

24

[C#] protected SoapExtension(); [C++] protected: SoapExtension(); [VB] Protected Sub New()

[JScript] protected function SoapExtension();

ChainStream

[C#] public virtual Stream ChainStream(Stream stream);

[C++] public: virtual Stream\* ChainStream(Stream\* stream);

[VB] Overridable Public Function ChainStream(ByVal stream As Stream) As Stream

[JScript] public function ChainStream(stream : Stream) : Stream;

# Description

When overridden in a derived class, allows a SOAP extension access to the memory buffer containing the SOAP request or response.

Return Value: A System.IO.Stream representing a new memory buffer that this SOAP extension can modify.

System.Web.Services.Protocols.SoapExtension.ChainStream(System.I **O.Stream**) ensures that SOAP extensions with the highest priority can modify the actual data closest to the SOAP message sent or returned over the wire. For instance, you probably would not want a compression SOAP extension to compress the data prior to an encryption SOAP extension encrypted the data. Doing so might lead to an inefficiently compressed data packet sent over the wire.

A memory buffer containing the SOAP request or response.

8

H 1 15 ļ. **4.** 17

18 19

20 21

22 23

24

[C#] public abstract object GetInitializer(Type serviceType);

[C++] public: virtual Object\* GetInitializer(Type\* serviceType) = 0;

[VB] MustOverride Public Function GetInitializer(ByVal serviceType As Type) As Object

[JScript] public abstract function GetInitializer(serviceType: Type): Object;

Description

When overridden in a derived class, allows a SOAP extension to initialize data specific to a class implementing a Web Service at a one-time performance cost.

Return Value: The System. Object that the SOAP extension initializes to be cached.

The overload of

System.Web.Services.Protocols.SoapExtension.GetIntializer that gets called by the ASP.NET Web Services infrastructure depends on how the SOAP extension was specified. There are two methods for specifying a SOAP extension: apply a custom attribute, deriving from

System. Web. Services. Protocols. Soap Extension Attribute, to the individual Web Service method or add a reference in either the web.config or app.config configuration files. If you add a reference to one of the configuration files, the SOAP extension runs for all Web Services within the scope of that configuration file. The ASP.NET Web Services infrastructure invokes the

System. Web. Services. Protocols. GetInitializer overload that passes in a

3

18 19

20

22

24

25

System.Type; otherwise the ASP.NET Web Services infrastructure invokes the System.Web.Services.Protocols.SoapExtension.GetInitializer(System.Web.Services.Protocols.LogicalMethodInfo,System.Web.Services.Protocols.SoapExtensionAttribute) that passes in a

System.Web.Services.Protocols.LogicalMethodInfo and a

**System.Web.Services.Protocols.SoapExtensionAttribute** . The type of the class implementing the Web Service that a SOAP extension is applied to.

GetInitializer

[C#] public abstract object GetInitializer(LogicalMethodInfo methodInfo, SoapExtensionAttribute attribute);

[C++] public: virtual Object\* GetInitializer(LogicalMethodInfo\* methodInfo, SoapExtensionAttribute\* attribute) = 0;

[VB] MustOverride Public Function GetInitializer(ByVal methodInfo As LogicalMethodInfo, ByVal attribute As SoapExtensionAttribute) As Object [JScript] public abstract function GetInitializer(methodInfo: LogicalMethodInfo, attribute: SoapExtensionAttribute): Object; When overridden in a derived class, allows a SOAP extension to initialize data specific to a Web Service method at a one-time performance cost.

#### Description

When overridden in a derived class, allows a SOAP extension to initialize data specific to a Web Service method using an attribute applied to the Web Service method at a one-time performance cost.

2

3

21

18

19

Return Value: The System.Object that the SOAP extension initializes to be cached.

If the SOAP extension is configured using a configuration file see the System. Web. Services. Protocols. Soap Extension. GetInitialzier overload that  $accepts\ a\ System. We b. Services. Protocols. Logical Method Info$ object representing the specific function prototype for the Web Service method the SOAP extension is applied to. The instance of the

System. Web. Services. Protocols. Soap Extension Attribute applied to the Web Service method.

Initialize

[C#] public abstract void Initialize(object initializer);

[C++] public: virtual void Initialize(Object\* initializer) = 0;

[VB] MustOverride Public Sub Initialize(ByVal initializer As Object)

[JScript] public abstract function Initialize(initializer : Object);

## Description

When overridden in a derived class, allows a SOAP extension to initialize itself using the data cached in the

System. Web. Services. Protocols. Soap Extension. Get Initializer (System. Web. Services) and the protocol of the protocol orvices. Protocols. Logical Method Info, System. Web. Services. Protocols. Soap Extervices and the protocol of the protocol onsionAttribute) method.

A SOAP extension has three opportunities to initialize data and they all have different purposes: Class constructor - The class constructor is called every time a SOAP extension is instantiated and is typically used to initialize member

MS1-863US.APP

21

22

23

24

variables. The System.Object returned from

System.Web.Services.Protocols.SoapExtension.GetInitializer(System.Web.Services.Protocols.LogicalMethodInfo,System.Web.Services.Protocols.SoapExtensionAttribute) cached by ASP.NET Web Services.

ProcessMessage

[C#] public abstract void ProcessMessage(SoapMessage message);

[C++] public: virtual void ProcessMessage(SoapMessage\* message) = 0;

[VB] MustOverride Public Sub ProcessMessage(ByVal message As

SoapMessage)

[JScript] public abstract function ProcessMessage(message : SoapMessage);

Description

When overridden in a derived class, allows a SOAP extension to receive a System.Web.Services.Protocols.SoapMessage to process at each System.Web.Services.Protocols.SoapMessageStage.

System.Web.Services.Protocols.SoapMessage) is called at all

System.Web.Services.Protocols.SoapMessageStage stages for SOAP extensions applied to both ASP.NET Web Service clients and ASP.NET Web Services. At each System.Web.Services.Protocols.SoapMessageStage, an instance of a class deriving from System.Web.Services.Protocols.SoapMessage is passed to System.Web.Services.Protocols.SoapExtension.ProcessMessage(System.Web. Services.Protocols.SoapMessage). If the SOAP extension is running on the Web Service client, then a System.Web.Services.Protocols.SoapClientMessage object

25

is passed into

System. Web. Services. Protocols. Soap Extension. Process Message (System. Web. System. Web. S

Services.Protocols.SoapMessage); otherwise a

System.Web.Services.Protocols.SoapServerMessage object is passed in. The System.Web.Services.Protocols.SoapMessage to process.

SoapExtensionAttribute class (System.Web.Services.Protocols)

**ToString** 

Description

When overridden in a derived class, specifies the SOAP extension should be applied to a Web Service method.

ASP.NET Web Services enables applying SOAP extensions to a Web Service method by applying an attribute. When a custom extension attribute is added to a Web Service method or a proxy class client, ASP.NET Web Services invokes the associated extension at the appropriate time. An extension attribute is a custom attribute class deriving from

System.Web.Services.Protocols.SoapExtensionAttribute . Derived attributes must override the

System.Web.Services.Protocols.SoapExtensionAttribute.ExtensionType property to return the type of extension that is associated with the attribute.

SoapExtensionAttribute

Example Syntax:

**ToString** 

```
[C#] protected SoapExtensionAttribute();
       [C++] protected: SoapExtensionAttribute();
       [VB] Protected Sub New()
       [JScript] protected function SoapExtensionAttribute();
              ExtensionType
   6
              ToString
   7
   8
       [C#] public abstract Type ExtensionType {get;}
       [C++] public: __property virtual Type* get ExtensionType() = 0;
10
11
11
12
12
       [VB] MustOverride Public ReadOnly Property ExtensionType As Type
       [JScript] public abstract function get ExtensionType(): Type;
       Description
              When overridden in a derived class, gets the System. Type of the SOAP
       extension.
Derived classes must override the
i= 17
       System. Web. Services. Protocols. Soap Extension Attribute. Extension Type \\
  18
       property to return the System.Type of SOAP extension.
  19
              Priority
  20
              ToString
  21
  22
      [C#] public abstract int Priority {get; set;}
  23
      [C++] public: __property virtual int get_Priority() = 0;public: __property virtual
      void set_Priority(int) = 0;
```

22

23

24

25

[VB] MustOverride Public Property Priority As Integer

[JScript] public abstract function get Priority(): int;public abstract function set Priority(int);

Description

When overridden in a derived class, gets or set the priority of the SOAP extension.

TypeId

SoapHeader class (System.Web.Services.Protocols)

**ToString** 

## Description

When overridden in a derived class, represents the content of a SOAP header.

SOAP headers offer a method for passing data to and from a Web Service method not directly related to a Web Service method's primary functionality. For instance, a Web Service may contain several Web Service methods that each require a custom authentication scheme. Instead of adding parameters to each Web Service method for the custom authentication scheme, a 
System.Web.Services.Protocols.SoapHeaderAttribute, referring to a class deriving from System.Web.Services.Protocols.SoapHeader, can be applied to each Web Service method. The implementation for the class deriving from System.Web.Services.Protocols.SoapHeader handles the custom authentication

scheme. In this manner, the Web Service method implements only the

```
functionality specific to it and adds additional functionality through the use of a
   1
       SOAP header.
   2
              SoapHeader
   3
              Example Syntax:
              ToString
   5
   6
       [C#] protected SoapHeader();
   7
       [C++] protected: SoapHeader();
   8
       [VB] Protected Sub New()
   9
       [JScript] protected function SoapHeader();
10
11
11
12
12
13
              Actor
              ToString
III WING THE
       [C#] public string Actor {get; set;}
       [C++] public: __property String* get Actor();public: __property void
set Actor(String*);
[VB] Public Property Actor As String
       [JScript] public function get Actor(): String; public function set Actor(String);
  18
  19
       Description
  20
              Gets or sets the recipient of the SOAP header.
  21
              According to the SOAP specification, the recipient (commonly called the
  22
       SOAP actor attribute) does not have to be the same for the Header and Body
  23
       elements of a SOAP message. For a Web Service method, the required Body
  24
```

rdi

25

element of a SOAP message represents the parameters and return values of the

20

21

22

23

24

25

1

2

Web Service method. If a **Header** element exists in the SOAP message, it represents additional data that can be sent to and from the Web Service method. The recipient of that data, known as the **Actor**, can be a different URI than the URI for the Web Service method.

DidUnderstand

**ToString** 

[C#] public bool DidUnderstand {get; set;}

[C++] public: \_\_property bool get\_DidUnderstand();public: \_\_property void set DidUnderstand(bool);

[VB] Public Property DidUnderstand As Boolean

[JScript] public function get DidUnderstand() : Boolean; public function set DidUnderstand(Boolean);

## Description

Gets or sets a value indicating whether a Web Service method properly processed a SOAP header.

For SOAP headers defined by a Web Service, ASP.NET Web Services assumes the Web Service method properly processed the SOAP header by setting the initial value of

System.Web.Services.Protocols.SoapHeader.DidUnderstand to true . For SOAP headers not defined by the Web Service, the initial value is false . If ASP.NET Web Services detects SOAP headers passed to a Web Service method with System.Web.Services.Protocols.SoapHeader.DidUnderstand set to false after the method returns, a

1	System.Web.Services.Protocols.SoapHeaderException is thrown back to the
2	Web Service client instead of the results from the Web Service method.
3	EncodedMustUnderstand
4	ToString
5	
6	[C#] public string EncodedMustUnderstand {get; set;}
7	[C++] public:property String* get_EncodedMustUnderstand();public:
8	property void set_EncodedMustUnderstand(String*);
9	[VB] Public Property EncodedMustUnderstand As String
10	[JScript] public function get EncodedMustUnderstand(): String;public function
10	set EncodedMustUnderstand(String);
12	
13 13 13 L	Description
l.	Gets or sets the value of the <b>mustUnderstand</b> attribute for the SOAP
14   14   15	header.
16	Valid values for the
17	System.Web.Services.Protocols.SoapHeader.EncodedMustUnderstand
18	property are 0 and 1, whereas setting the property to false or true equates to setting
19	the property to 0 or 1, respectively.
20	MustUnderstand
21	ToString
22	
23	[C#] public bool MustUnderstand {get; set;}
24	[C++] public:property bool get_MustUnderstand();public:property void
25	set_MustUnderstand(bool);

24

25

[VB] Public Property MustUnderstand As Boolean

[JScript] public function get MustUnderstand() : Boolean; public function set MustUnderstand(Boolean);

### Description

Gets or sets a value indicating whether the

System. Web. Services. Protocols. Soap Header must be understood.

When a Web Service client adds a SOAP header to a Web Service method call with the System.Web.Services.Protocols.SoapHeader.MustUnderstand property set to true, the Web Service method must set the

System.Web.Services.Protocols.SoapHeader.DidUnderstand property to true; otherwise, a System.Web.Services.Protocols.SoapHeaderException is thrown back to the Web Service client by ASP.NET Web Services.

SoapHeaderAttribute class (System.Web.Services.Protocols)
ToString

## Description

This attribute is applied to a Web Service or a Web Service client to specify a SOAP Header the Web Service or Web Service client wants to process. This class cannot be inherited.

The basic steps to receiving and processing a SOAP header are: Create a class deriving from **System.Web.Services.Protocols.SoapHeader** representing the data passed in the SOAP header.

SoapHeaderAttribute

3

5 6

7

8

9

[]10

111

17

19

20

18

21

22

23

24

25

Example Syntax: **ToString** 

[C#] public SoapHeaderAttribute(string memberName);

[C++] public: SoapHeaderAttribute(String\* memberName);

[VB] Public Sub New(ByVal memberName As String)

[JScript] public function SoapHeaderAttribute(memberName : String);

Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapHeaderAttribute class setting the member of the Web Service class representing the SOAP header contents.

A Web Service client or Web Service can specify the SOAP headers it wants to process for specific Web Service methods by applying a System.Web.Services.Protocols.SoapHeaderAttribute to the Web Service method in the Web Service class or the corresponding method in the proxy class. In order for the Web Service method to receive the contents of the SOAP header, a member is added to the Web Service class of a System. Type derived from System. Web. Services. Protocols. Soap Header. Likewise, a Web Service client adds a member to the proxy class deriving from System.Web.Services.Protocols.SoapHeader . That System.Type deriving from System. Web. Services. Protocols. Soap Header represents the contents of the SOAP header and is created by the developer of the Web Service or Web Service client that wants to receive the SOAP header. Once the class is created, a System. Web. Services. Protocols. Soap Header Attribute can be applied to the

1	Web Service method or corresponding method in the Web Service client proxy
2	class specifing the member to receive the SOAP header contents with the
3	memberName parameter to this constructor or the
4	System.Web.Services.Protocols.SoapHeaderAttribute.MemberName property.
5	Member of the Web Service class representing the SOAP header contents. Sets the
6	System.Web.Services.Protocols.SoapHeaderAttribute.MemberName property.
7	Direction
8	ToString
9	
<b>=</b> 10	[C#] public SoapHeaderDirection Direction {get; set;}
[]10 []11	[C++] public:property SoapHeaderDirection get_Direction();public:property
12 12	void set_Direction(SoapHeaderDirection);
13   13	[VB] Public Property Direction As SoapHeaderDirection
	[JScript] public function get Direction(): SoapHeaderDirection; public function set
14	Direction(SoapHeaderDirection);
16	
17	Description
18	Gets or sets whether the SOAP header is intended for the Web Service or
19	the Web Service client or both.
20	MemberName
21	ToString
22	
23	[C#] public string MemberName {get; set;}
24	[C++] public:property String* get_MemberName();public:property void
25	set_MemberName(String*);

9

7

17

19

[VB] Public Property MemberName As String

[JScript] public function get MemberName(): String; public function set

MemberName(String);

Description

Gets or sets the member of the Web Service class representing the SOAP header contents.

A Web Service client or Web Service can specify the SOAP headers it wants to process for specific Web Service methods by applying a System.Web.Services.Protocols.SoapHeaderAttribute to the Web Service method in the Web Service class or the corresponding method in the proxy class. In order for the Web Service method to receive the contents of the SOAP header, a member is added to the Web Service class of a System. Type derived from System. Web. Services. Protocols. Soap Header. Likewise, a Web Service client adds a member to the proxy class deriving from  ${\bf System. Web. Services. Protocols. Soap Header}\ .\ {\bf That}\ {\bf System. Type}\ deriving\ from$ System. Web. Services. Protocols. Soap Header represents the contents of the SOAP header and is created by the developer of the Web Service or Web Service client that wants to receive the SOAP header. Once the class is created, a System. Web. Services. Protocols. Soap Header Attribute can be applied to the Web Service method or corresponding method in the Web Service client proxy class specifing the member to receive the SOAP header contents with the  $System. Web. Services. Protocols. Soap Header Attribute. Member Name\ property.$ 

Required

ToString

ı	
2	[C#] public bool Required {get; set;}
,	[C++] public:property bool get_Required();public:property void
	set_Required(bool);
5	[VB] Public Property Required As Boolean
5	[JScript] public function get Required(): Boolean; public function set
,	Required(Boolean);
3	
,	Description
,	Gets or sets a value indicating whether the SOAP header must be
	understood and processed by the recipient Web Service or Web Service client.
2	When the System. Type of the member (represented by the
3	System.Web.Services.Protocols.SoapHeaderAttribute.MemberName property)
4	is either System.Web.Services.Protocols.SoapHeader or
5	System.Web.Services.Protocols.SoapUnknownHeader, the
5	System.Web.Services.Protocols.SoapHeaderAttribute.Required property must
,	be <b>false</b> . For types that derive from
3	System.Web.Services.Protocols.SoapHeader, it is fine for the
9	System.Web.Services.Protocols.SoapHeaderAttribute.Required property to be
	true or false .
1	TypeId
2	SoapHeaderCollection class (System.Web.Services.Protocols)
3	ToString
4	

17

19 20

22

21

24

25

23

Description

Contains a collection of **System.Web.Services.Protocols.SoapHeader** objects.

SoapHeaderCollection

Example Syntax:

**ToString** 

[C#] public SoapHeaderCollection();

[C++] public: SoapHeaderCollection();

[VB] Public Sub New()

[JScript] public function SoapHeaderCollection();

Count

InnerList

Item

**ToString** 

Description

Gets or sets the **System.Web.Services.Protocols.SoapHeader** at position *index* of the **System.Web.Services.Protocols.SoapHeaderCollection** .

This property provides the ability to access a specific element in the collection by using the following syntax: myCollection[index]. The zero-based index of the System.Web.Services.Protocols.SoapHeader to get or set.

List Add 3 [C#] public int Add(SoapHeader header); [C++] public: int Add(SoapHeader\* header); [VB] Public Function Add(ByVal header As SoapHeader) As Integer [JScript] public function Add(header : SoapHeader) : int; 8 Description Adds a System. Web. Services. Protocols. Soap Header to the 10 System. Web. Services. Protocols. Soap Header Collection. Return Value: The position into which the 12 System.Web.Services.Protocols.SoapHeader was inserted. The 13 System.Web.Services.Protocols.SoapHeader to add to the 14 System. Web. Services. Protocols. Soap Header Collection. 15 Contains 16 17 [C#] public bool Contains(SoapHeader header); 18 [C++] public: bool Contains(SoapHeader\* header); 19 [VB] Public Function Contains(ByVal header As SoapHeader) As Boolean 20 [JScript] public function Contains(header : SoapHeader) : Boolean; 21 22 Description 23 Determines whether the 24 System.Web.Services.Protocols.SoapHeaderCollection contains a specific

1	System.Web.Services.Protocols.SoapHeader.
2	Return Value: true if the System. Web. Services. Protocols. Soap Header is found
3	in the System. Web. Services. Protocols. Soap Header Collection; otherwise, falso
4	. The System. Web. Services. Protocols. Soap Header to locate in the
5	System.Web.Services.Protocols.SoapHeaderCollection.
6	СоруТо
7	
8	[C#] public void CopyTo(SoapHeader[] array, int index);
9	[C++] public: void CopyTo(SoapHeader* array[], int index);
10	[VB] Public Sub CopyTo(ByVal array() As SoapHeader, ByVal index As Integer
11	[JScript] public function CopyTo(array : SoapHeader[], index : int);
12	
13	Description
14	Copies the elements of the
15	System.Web.Services.Protocols.SoapHeaderCollection to an System.Array,
16	starting at a particular System.Array index. The one-dimensional System.Array
17	that is the destination of the elements copied from
18	System.Web.Services.Protocols.SoapHeaderCollection. The System.Array
19	must have zero-based indexing. The zero-based index in array at which copying
20	begins.
21	IndexOf
22	
23	[C#] public int IndexOf(SoapHeader header);
24	[C++] public: int IndexOf(SoapHeader* header);
25	[VB] Public Function IndexOf(ByVal header As SoapHeader) As Integer

[JScript] public function IndexOf(header: SoapHeader): int;

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Determines the index of the SoapHeader in the SoapHeaderCollection.

Return Value: The index of header if found in the

 ${\bf System. Web. Services. Protocols. Soap Header Collection}\ ; \ otherwise, -1.\ The$ 

System.Web.Services.Protocols.SoapHeaderto locate in the

System. Web. Services. Protocols. Soap Header Collection.

Insert

[C#] public void Insert(int index, SoapHeader header);

[C++] public: void Insert(int index, SoapHeader\* header);

[VB] Public Sub Insert(ByVal index As Integer, ByVal header As SoapHeader)

[JScript] public function Insert(index : int, header : SoapHeader);

Description

Inserts a System.Web.Services.Protocols.SoapHeader into the System.Web.Services.Protocols.SoapHeaderCollection at the specified index.

If index equals the number of items in the collection, then the System.Web.Services.Protocols.SoapHeader is appended to the end. The zero-based index at which the System.Web.Services.Protocols.SoapHeader is inserted into the System.Web.Services.Protocols.SoapHeaderCollection . The System.Web.Services.Protocols.SoapHeader inserted into the System.Web.Services.Protocols.SoapHeaderCollection .

Remove

[C#] public void Remove(SoapHeader header);[C++] public: void Remove(SoapHeader\* header);[VB] Public Sub Remove(ByVal header As SoapHeader)[JScript] public function Remove(header: SoapHeader);

### Description

Removes the first occurrence of a specific System.Web.Services.Protocols.SoapHeader from the System.Web.Services.Protocols.SoapHeaderCollection .

The **System.Web.Services.Protocols.SoapHeader** elements that follow the removed **System.Web.Services.Protocols.SoapHeader** move up to occupy the vacated spot. The **System.Web.Services.Protocols.SoapHeader** to remove from the **System.Web.Services.Protocols.SoapHeaderCollection**.

SoapHeaderDirection enumeration (System.Web.Services.Protocols)
ToString

#### Description

Specifies whether the recipient of the

System.Web.Services.Protocols.SoapHeader is the Web Service, the Web

Service client or both.

A System.Web.Services.Protocols.SoapHeaderAttribute can be applied to either a Web Service method or a method on a proxy class to a Web Service. In either case, the recipient is specified by the

	1	System.Web.Services.Protocols.SoapHeaderAttribute.Direction property of
	2	System.Web.Services.Protocols.SoapHeaderAttribute .
	3	ToString
	4	
	5	[C#] public const SoapHeaderDirection In;
	6	[C++] public: const SoapHeaderDirection In;
	7	[VB] Public Const In As SoapHeaderDirection
	8	[JScript] public var In : SoapHeaderDirection;
	9	
	10	Description
	11	Specifies the System. Web. Services. Protocols. Soap Header is sent to the
	12	Web Service.
	13	ToString
	14	
### ## ###############################	15	[C#] public const SoapHeaderDirection InOut;
Han	16	[C++] public: const SoapHeaderDirection InOut;
ulla	17	[VB] Public Const InOut As SoapHeaderDirection
	18	[JScript] public var InOut : SoapHeaderDirection;
	19	
	20	Description
	21	Specifies the System. Web. Services. Protocols. Soap Header is sent to both
	22	the Web Service and the Web Service client.
	23	ToString
	24	
	25	[C#] public const SoapHeaderDirection Out;

20

21

22

23

24

25

1	[C++] public: const SoapHeaderDirection Out;
2	[VB] Public Const Out As SoapHeaderDirection
3	[JScript] public var Out : SoapHeaderDirection
4	
5	Description
6	Specifies the System.Web.Services.Pro
7	Web Service client.
8	SoapHeaderException class (System.We
9	ToString
10	
11	
12	Description
13	The exception that is thrown when a W
14	Simple Object Access Protocol (SOAP) and an
15	SOAP header.
16	When a Web Service client adds a SOA
17	call with the System.Web.Services.Protocols
18	property set to <b>true</b> , the Web Service method

vices.Protocols.SoapHeader is sent to the

ystem.Web.Services.Protocols)

when a Web Service method is called over P) and an exception occurs processing the

lds a SOAP header to a Web Service method Protocols.SoapHeader.MustUnderstand e method must set the  ${\bf System. Web. Services. Protocols. Soap Header. Did Understand \ property \ to}$ true; otherwise a System. Web. Services. Protocols. Soap Header Exception is thrown.

SoapHeaderException

Example Syntax:

**ToString** 

1	
2	[C#] public SoapHeaderException(string message, XmlQualifiedName code);
3	[C++] public: SoapHeaderException(String* message, XmlQualifiedName*
4	code);
5	[VB] Public Sub New(ByVal message As String, ByVal code As
6	XmlQualifiedName)
7	[JScript] public function SoapHeaderException(message : String, code :
8	XmlQualifiedName);
9	
10	Description
-11	Initializes a new instance of the
12	System.Web.Services.Protocols.SoapHeaderException class. A message that
13	identifies the reason the exception occurred. Sets the System.Exception.Message
14	property. The type of error that occurred. Sets the
15	System.Web.Services.Protocols.SoapException.Code property.
16	SoapHeaderException
17	Example Syntax:
18	ToString
19	
20	[C#] public SoapHeaderException(string message, XmlQualifiedName code,
21	Exception innerException);
22	[C++] public: SoapHeaderException(String* message, XmlQualifiedName* code,
23	Exception* innerException);
24	[VB] Public Sub New(ByVal message As String, ByVal code As
25	XmlQualifiedName, ByVal innerException As Exception)

[JScript] public function SoapHeaderException(message: String, code: XmlQualifiedName, innerException: Exception); 3 Description Initializes a new instance of the 5 System.Web.Services.Protocols.SoapHeaderException class. A message that identifies the reason the exception occurred. Sets the System. Exception. Message 7 property. The type of error that occurred. Sets the 8 System. Web. Services. Protocols. Soap Exception. Code property. A reference to the root cause of an exception. Sets the System. Exception. Inner Exception 10 property. 11 SoapHeaderException 12 Example Syntax: 13 **ToString** 14 15 [C#] public SoapHeaderException(string message, XmlQualifiedName code, 16 string actor); 17 [C++] public: SoapHeaderException(String\* message, XmlQualifiedName\* code, 18 String\* actor); 19 [VB] Public Sub New(ByVal message As String, ByVal code As 20 XmlQualifiedName, ByVal actor As String) 21 [JScript] public function SoapHeaderException(message : String, code : 22 XmlQualifiedName, actor: String); 23 24 Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapHeaderException class. A message that
identifies the reason the exception occurred. Sets the System.Exception.Message
property. The type of error that occurred. Sets the

System.Web.Services.Protocols.SoapException.Code property. The piece of
code that caused the exception. Typically, a URL to a Web Service method. Sets
the System.Web.Services.Protocols.SoapException.Actor property.

SoapHeaderException

Example Syntax:

**ToString** 

[C#] public SoapHeaderException(string message, XmlQualifiedName code, string actor, Exception innerException);

[C++] public: SoapHeaderException(String\* message, XmlQualifiedName\* code, String\* actor, Exception\* innerException);

[VB] Public Sub New(ByVal message As String, ByVal code As

XmlQualifiedName, ByVal actor As String, ByVal innerException As Exception)

[JScript] public function SoapHeaderException(message : String, code :

XmlQualifiedName, actor: String, innerException: Exception);

Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapHeaderException class. A message that identifies the reason the exception occurred. Sets the System.Exception.Message property. The type of error that occurred. Sets the

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

System. Web. Services. Protocols. Soap Exception. Code property. The piece of code that caused the exception. Typically, a URL to a Web Service method. Sets the System. Web. Services. Protocols. Soap Exception. Actor property. A reference to the root cause of an exception. Sets the System. Exception. Inner Exception property. SoapHeaderException Example Syntax: **ToString** [C#] public SoapHeaderException(string message, XmlQualifiedName code, string actor, XmlNode[] otherElements);

[C++] public: SoapHeaderException(String\* message, XmlQualifiedName\* code, String\* actor, XmlNode\* otherElements[]);

[VB] Public Sub New(ByVal message As String, ByVal code As

XmlQualifiedName, ByVal actor As String, ByVal otherElements() As XmlNode)

[JScript] public function SoapHeaderException(message : String, code :

 $XmlQualifiedName,\ actor: String,\ other Elements: XmlNode[]);$ 

# Description

Initializes a new instance of the

System.Web.Services.Protocols.SoapHeaderException class. A message that identifies the reason the exception occurred. Sets the System. Exception. Message property. The type of error that occurred. Sets the

 ${\bf System. Web. Services. Protocols. Soap Exception. Code}\ {\bf property.}\ {\bf The}\ {\bf piece}\ {\bf of}$ code that caused the exception. Typically, a URL to a Web Service method. Sets

the System.Web.Services.Protocols.SoapException.Actor property. Optional 1 exception information. Sets the 2  ${\bf System. Web. Services. Protocols. Soap Exception. Other Elements \ property.}$ 3 SoapHeaderException 4 Example Syntax: 5 **ToString** 6 7 [C#] public SoapHeaderException(string message, XmlQualifiedName code, 8 string actor, XmlNode[] otherElements, Exception innerException); 9 [C++] public: SoapHeaderException(String\* message, XmlQualifiedName\* code, 10 String\* actor, XmlNode\* otherElements[], Exception\* innerException); 11 [VB] Public Sub New(ByVal message As String, ByVal code As 12 XmlQualifiedName, ByVal actor As String, ByVal otherElements() As XmlNode, 13 ByVal innerException As Exception) 14 [JScript] public function SoapHeaderException(message: String, code: 15 XmlQualifiedName, actor: String, otherElements: XmlNode[], innerException: 16 Exception); 17 18 Description 19 Initializes a new instance of the 20 System.Web.Services.Protocols.SoapHeaderException class. A message that 21 identifies the reason the exception occurred. Sets the System. Exception. Message 22 property. The type of error that occurred. Sets the 23 System.Web.Services.Protocols.SoapException.Code property. The piece of 24 code that caused the exception. Typically, a URL to a Web Service method. Sets

the System. Web. Services. Protocols. Soap Exception. Actor property. Optional exception information. Sets the 2  ${\bf System. Web. Services. Protocols. Soap Exception. Other Elements\ property.\ A}$ 3 reference to the root cause of an exception. Sets the 4 System. Exception. Inner Exception property. 5 Actor Code 7 Detail 8 HelpLink 9 **HResult** 10 InnerException 11 Message 12 OtherElements 13 Source 14 StackTrace 15 **TargetSite** 16 SoapHttpClientProtocol class (System.Web.Services.Protocols) 17 **ToString** 18 19 20 Description 21 Specifies the class ASP.NET Web Service client proxies derive from when 22 using Simple Object Access Protocol (SOAP). 23

ASP.NET Web Services and Building Web Services clients. If you are building a

ASP.NET incorporates two distinct Web Services functionalities: Building

1	Web Service client using ASP.NET, then a proxy class deriving indirectly or
2	directly from System.Web.Services.Protocols.WebClientProtocol needs to be
3	created for the Web Service you want to call. When the Web Service client is
4	calling via SOAP, the proxy class should derive from
5	${\bf System. Web. Services. Protocols. Soap Http Client Protocol}\ , \ which\ derives\ from$
6	$System. Web. Services. Protocols. Http Web Client Protocol \ .$
7	SoapHttpClientProtocol
8	Example Syntax:
9	ToString
10	
11	[C#] public SoapHttpClientProtocol();
12	[C++] public: SoapHttpClientProtocol();
13	[VB] Public Sub New()
14	[JScript] public function SoapHttpClientProtocol();
15	
16	Description
17	Initializes a new instance of the
18	System.Web.Services.Protocols.SoapHttpClientProtocol class.
19	AllowAutoRedirect
20	ClientCertificates
21	ConnectionGroupName
22	Container
23	CookieContainer
24	Credentials
25	DesignMode

Events
PreAuthenticate
Proxy
RequestEncoding
Site
Timeout
Url
UserAgent
BeginInvoke

[C#] protected IAsyncResult BeginInvoke(string methodName, object[]
parameters, AsyncCallback callback, object asyncState);
[C++] protected: IAsyncResult\* BeginInvoke(String\* methodName, Object\*
parameters \_\_gc[], AsyncCallback\* callback, Object\* asyncState);
[VB] Protected Function BeginInvoke(ByVal methodName As String, ByVal
parameters() As Object, ByVal callback As AsyncCallback, ByVal asyncState As
Object) As IAsyncResult
[JScript] protected function BeginInvoke(methodName: String, parameters:
Object[], callback: AsyncCallback, asyncState: Object): IAsyncResult;

Description
Starts an asynchronous invocation of a Web Service method using Simple

Starts an asynchronous invocation of a Web Service method using Simple Object Access Protocol(SOAP).

Return Value: An System.IAsyncResult which is passed to

25

System. Web. Services. Protocols. Soap Http Client Protocol. End Invoke (System. Invoke Control of Control o**AsyncResult)** to obtain the return values from the remote method call. 2 Typically, you would not call the 3 System. Web. Services. Protocols. Soap Http Client Protocol. Begin Invoke (System. String, System. Object[], System. Async Callback, System. Object) method directly, unless you were building your own proxy class for a Web Service. The name of the Web Service method in the derived class that is invoking System. Web. Services. Protocols. Soap Http Client Protocol. Begin Invoke (System. 8 String, System. Object[], System. AsyncCallback, System. Object). An array of objects containing the parameters to pass to the remote web service. The order of 10 the values in the array correspond to the order of the parameters in the calling 11 method of the derived class. The delegate to call when the asynchronous invoke is complete. If callback is **null**, the delegate is not called. Extra information supplied 13 by the caller. 14 Discover 15 16 [C#] public void Discover(); 17 [C++] public: void Discover(); 18 [VB] Public Sub Discover() 19 [JScript] public function Discover(); 20 21 Description 22 Dynamically binds to a Web Service described in the discovery document 23

lee⊗hayes pilc 509-324-9256 910 MS1-863US.APP

at System. Web. Services. Protocols. Web Client Protocol. Url.

A Web Service client can dynamically bind to a Web Service other than the one referenced in the proxy class using the 
System.Web.Services.Protocols.SoapHttpClientProtocol.Discover method.

Typically, the System.Web.Services.Protocols.WebClientProtocol.Url property refers to the base address of the Web Service, however prior to invoking the 
System.Web.Services.Protocols.SoapHttpClientProtocol.Discover method, set the System.Web.Services.Protocols.WebClientProtocol.Url property to a URL of a discovery document. Invoking the

System.Web.Services.Protocols.SoapHttpClientProtocol.Discover method attempts to find a match in the discovery document to the binding defined in the proxy class and then dynamically bind to it. If successful, future method invocations are directed to the Web Service described in the discovery document.

EndInvoke

[C#] protected object[] EndInvoke(IAsyncResult asyncResult);
[C++] protected: Object\* EndInvoke(IAsyncResult\* asyncResult) \_\_gc[];
[VB] Protected Function EndInvoke(ByVal asyncResult As IAsyncResult) As
Object()

[JScript] protected function EndInvoke(asyncResult : IAsyncResult) : Object[];

Description

Ends an asynchronous invocation of a Web Service method using Simple Object Access Protocol(SOAP).

Return Value: An array of objects containing the return value and any byreference or out parameters of the derived class method.

21

22

23

1

Typically, you would not call the

System.Web.Services.Protocols.SoapHttpClientProtocol.EndInvoke(System.I

AsyncResult) method directly, unless you were building your own proxy class for a Web Service. The System.IAsyncResult returned from

System.Web.Services.Protocols.SoapHttpClientProtocol.BeginInvoke(System. String,System.Object[],System.AsyncCallback,System.Object).

GetWebRequest

[C#] protected override WebRequest GetWebRequest(Uri uri);

[C++] protected: WebRequest\* GetWebRequest(Uri\* uri);

[VB] Overrides Protected Function GetWebRequest(ByVal uri As Uri) As

WebRequest

[JScript] protected override function GetWebRequest(uri : Uri) : WebRequest;

Description

Creates a System.Net.WebRequest instance for the specified url.

Return Value: The System.Net.WebRequest instance.

This method can be overridden to customize the **System.Net.WebRequest** object before the Web Service request is made. For example you could add a custom header to the request. The **System.Uri** to use when creating the **System.Net.WebRequest**.

Invoke

[C#] protected object[] Invoke(string methodName, object[] parameters);

[C++] protected: Object\* Invoke(String\* methodName, Object\* parameters

\_gc[]) \_\_gc[]; 1 [VB] Protected Function Invoke(ByVal methodName As String, ByVal 2 parameters() As Object() As Object() 3 [JScript] protected function Invoke(methodName : String, parameters : Object[]) : 4 Object[]; 5 6 Description 7 Invokes a Web Service method synchronously using SOAP. 8 Return Value: An array of objects containing the return value and any by reference 9 or out parameters of the derived class method. 10 Typically, you would not call the 11 System.Web.Services.Protocols.SoapHttpClientProtocol.Invoke(System.Strin 12 g,System.Object[]) method directly, unless you were building your own proxy 13 class for a Web Service. The name of the Web Service method. An array of 14 objects containing the parameters to pass to the remote web service. The order of 15 the values in the array correspond to the order of the parameters in the calling 16 method of the derived class. 17 SoapMessage class (System.Web.Services.Protocols) 18 **ToString** 19 20 21 Description 22 Represents the data in a SOAP request or SOAP response at a specific 23 System.Web.Services.Protocols.SoapMessageStage . 24

lee@hayes oilc 509-324-9256 913 MS1-863US.APP

25

1	The System.Web.Services.Protocols.SoapMessage class is primary used
2	for SOAP extensions to represent the data in a SOAP request or SOAP response at
3	the System.Web.Services.Protocols.SoapMessageStage in the
4	System.Web.Services.Protocols.SoapMessage.Stage property. A SOAP
5	extension, which is a class deriving from
6	System.Web.Services.Protocols.SoapExtension, receives a
7	System.Web.Services.Protocols.SoapMessage at each
8	System.Web.Services.Protocols.SoapMessageStage when the
9	System. Web. Services. Protocols. Soap Extension. Process Message (System. Web. And System) and System and S
10	Services.Protocols.SoapMessage) method is called. It is up to the particular
11	SOAP extension to decide how to process the
12	System.Web.Services.Protocols.SoapMessage, but a couple of common SOAP
13	extensions might include encryption and compression.
14	Action
15	ToString
16	
17	[C#] public abstract string Action {get;}
18	[C++] public:property virtual String* get_Action() = 0;
19	[VB] MustOverride Public ReadOnly Property Action As String
20	[JScript] public abstract function get Action() : String;
21	
22	Description
23	When overridden in a derived class, gets the SOAPAction HTTP request

header field for the SOAP request or SOAP response.

1	The System.Web.Services.Protocols.SoapMessage.Action property can
2	be accessed during any System. Web. Services. Protocols. Soap Message Stage.
3	ContentType
4	ToString
5	
6	[C#] public string ContentType {get; set;}
7	[C++] public:property String* get_ContentType();public:property void
8	set_ContentType(String*);
9	[VB] Public Property ContentType As String
10	[JScript] public function get ContentType(): String; public function set
11	ContentType(String);
12	
13	Description
14	Gets or sets the HTTP Content-Type of the SOAP request or SOAP
15	response.
16	The System.Web.Services.Protocols.SoapMessage.ContentType
17	property can only be accessed in the
18	System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize and
19	System.Web.Services.Protocols.SoapMessageStage.BeforeDeserialize stages;
20	otherwise an System.InvalidOperationException is thrown.
21	Exception
22	ToString
23	
24	[C#] public SoapException Exception {get;}
25	[C++] public:property SoapException* get_Exception();

ToString

1	[VB] Public ReadOnly Property Exception As SoapException
2	[JScript] public function get Exception(): SoapException;
3	
4	Description
5	Gets the System.Web.Services.Protocols.SoapException from the call to
6	the Web Service method.
7	The System.Web.Services.Protocols.SoapMessage.Exception property
8	can be accessed during any System.Web.Services.Protocols.SoapMessageStage
9	•
10	Headers
11	ToString
12	
13	[C#] public SoapHeaderCollection Headers {get;}
14	[C++] public:property SoapHeaderCollection* get_Headers();
15	[VB] Public ReadOnly Property Headers As SoapHeaderCollection
16	[JScript] public function get Headers(): SoapHeaderCollection;
17	
18	Description
19	A collection of the SOAP headers applied to the current SOAP request or
20	SOAP response.
21	SOAP headers are an optional portion of a SOAP request or SOAP
22	response. For more details on using SOAP headers in ASP.NET Web Services and
23	ASP.NET clients, see the System.Web.Services.Protocols.SoapHeader class.
24	MethodInfo

1	
2	[C#] public abstract LogicalMethodInfo MethodInfo {get;}
3	[C++] public:property virtual LogicalMethodInfo* get_MethodInfo() = 0;
4	[VB] MustOverride Public ReadOnly Property MethodInfo As LogicalMethodInfo
5	[JScript] public abstract function get MethodInfo(): LogicalMethodInfo;
6	
7	Description
8	When overridden in a derived class, gets a representation of the method
9	prototype for the Web Service method for which the SOAP request is intended.
10	Although the System. Web. Services. Protocols. Soap Message. Method Info
11	can be accessed during any System. Web. Services. Protocols. Soap Message Stage
12	, it only has data during
13	System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize and
14	System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize .
15	OneWay
16	ToString
17	
18	[C#] public abstract bool OneWay {get;}
19	[C++] public:property virtual bool get_OneWay() = 0;
20	[VB] MustOverride Public ReadOnly Property OneWay As Boolean
21	[JScript] public abstract function get OneWay(): Boolean;
22	
23	Description
24	Gets a value indicating the
, ,	System, Web, Services, Protocols, Soan Document Method Attribute, One Way

property of either the  $System. Web. Services. Protocols. Soap Document Method Attribute \ or \\$ 2 System.Web.Services.Protocols.SoapRpcMethodAttribute applied to the Web 3 Service method. See the 5 System.Web.Services.Protocols.SoapDocumentMethodAttribute.OneWay 6 property of System. Web. Services. Protocols. Soap Document Method Attribute or 7 System.Web.Services.Protocols.SoapRpcMethodAttribute for details about 8 one-way Web Service methods. 9 Stage 10 **ToString** 11 12 [C#] public SoapMessageStage Stage {get;} 13 [C++] public: property SoapMessageStage get Stage(); 14 [VB] Public ReadOnly Property Stage As SoapMessageStage 15 [JScript] public function get Stage() : SoapMessageStage; 16 17 Description 18 Gets the System. Web. Services. Protocols. Soap Message Stage of the 19 System.Web.Services.Protocols.SoapMessage. 20 The System.Web.Services.Protocols.SoapMessage.Stage property can be 21 accessed at any System. Web. Services. Protocols. Soap Message Stage. 22 Stream 23 **ToString** 24 25

1 [C#] public Stream Stream {get;} [C++] public: property Stream\* get Stream(); 3 [VB] Public ReadOnly Property Stream As Stream [JScript] public function get Stream(): Stream; 5 6 Description 7 Gets the data representing the SOAP request or SOAP response in the form 8 of a System.IO.Stream object. 9 SOAP extensions that inspect the data representing a SOAP request or 10 SOAP response, such as a logging SOAP extension, can use the 11 System.IO.Stream property to view the data at each 12 System.Web.Services.Protocols.SoapMessageStage. 13 Url 14 **ToString** 15 16 [C#] public abstract string Url {get;} 17 [C++] public: property virtual String\* get Url() = 0; 18 [VB] MustOverride Public ReadOnly Property Url As String 19 [JScript] public abstract function get Url(): String; 20 21 Description 22 When overridden in a derived class, gets the base URL of the Web Service. 23 The System. Web. Services. Protocols. Soap Message. Url property can be 24

accessed during any System. Web. Services. Protocols. Soap Message Stage.

3

4

5

6

7

8

9

10

11 12

13

14 15

16 17

18

19

20

22

23

24

25

[C#] protected abstract void EnsureInStage();

[C++] protected: virtual void EnsureInStage() = 0;

[VB] MustOverride Protected Sub EnsureInStage()

[JScript] protected abstract function EnsureInStage();

Description

When overridden in a derived class, asserts that the current **System.Web.Services.Protocols.SoapMessageStage** stage is a stage where in parameters are available.

The asserted stage is different for Web Service clients and Web Service methods. A Web Service method's in parameters are available after the SOAP request is describilized in the

**System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize** stage. For a Web Service client, the in parameters are available prior to the SOAP request serialization process in the

 $System. Web. Services. Protocols. Soap Message Stage. Before Serialize \ {\tt stage}.$ 

EnsureOutStage

[C#] protected abstract void EnsureOutStage();

[C++] protected: virtual void EnsureOutStage() = 0;

[VB] MustOverride Protected Sub EnsureOutStage()

[JScript] protected abstract function EnsureOutStage();

16

17

18

19

20

21

22

23

### Description

1

2

3

4

5

When overridden in a derived class, asserts that the current **System.Web.Services.Protocols.SoapMessageStage** stage is a stage where out parameters are available.

The asserted stage is different for Web Service clients and Web Service methods. A Web Service method's out parameters are available prior to the SOAP response serialization process in the

System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize stage. For a Web Service client, the out parameters are available after the SOAP response describilization process in the

System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize stage.

EnsureStage

[C#] protected void EnsureStage(SoapMessageStage stage);

[C++] protected: void EnsureStage(SoapMessageStage stage);

[VB] Protected Sub EnsureStage(ByVal stage As SoapMessageStage)

[JScript] protected function EnsureStage(stage : SoapMessageStage);

# Description

Ensures that the **System.Web.Services.Protocols.SoapMessageStage** of the call to the Web Service method is the stage or stages passed in. If the current processing stage is not one of the stages passed in, an exception is thrown.

24

25

Multiple stages can be asserted by performing a bitwise OR operation on multiple System.Web.Services.Protocols.SoapMessageStage stages. The System.Web.Services.Protocols.SoapMessageStage asserted.

GetInParameterValue

[C#] public object GetInParameterValue(int index);

[C++] public: Object\* GetInParameterValue(int index);

[VB] Public Function GetInParameterValue(ByVal index As Integer) As Object [JScript] public function GetInParameterValue(index : int) : Object;

Description

Gets the parameter passed into the Web Service method at the specified index.

Return Value: An System.Object representing the parameter at index in the array of parameters.

The

System.Web.Services.Protocols.SoapMessage.GetInParameterValue(System.I nt32) property is only accessible when in parameters are available. ASP.NET Web Services throws System.InvalidOperationException when accessing the System.Web.Services.Protocols.SoapMessage.GetInParameterValue(System.I nt32) property if in parameters are not available. The zero-based index of the parameter in the array of parameters.

GetOutParameterValue

[C#] public object GetOutParameterValue(int index);

2

3

4

[C++] public: Object\* GetOutParameterValue(int index); [VB] Public Function GetOutParameterValue(ByVal index As Integer) As Object [JScript] public function GetOutParameterValue(index : int) : Object; Description 5 Gets the out parameter passed into the Web Service method at the specified 6 index. 7 Return Value: An System.Object representing the parameter at index in the array 8 of parameters. 9 The 10 System. Web. Services. Protocols. Soap Message. Get In Parameter Value (System. In Parameter Value) and the protocol of the11 nt32) property is only accessible when out parameters are available. ASP.NET 12 Web Services throws System.InvalidOperationException when accessing the 13 System.Web.Services.Protocols.SoapMessage.GetOutParameterValue(System 14 .Int32) property if out parameters are not available. The zero-based index of the 15 parameter in the array of parameters. 16 GetReturnValue 17 18 [C#] public object GetReturnValue(); 19 [C++] public: Object\* GetReturnValue(); 20 [VB] Public Function GetReturnValue() As Object 21 [JScript] public function GetReturnValue(): Object; 22 23 Description 24

Gets the return value of a Web Service method.

Return Value: An System.Object representing the return value of the Web Service method.

The System.Web.Services.Protocols.SoapMessage.GetReturnValue property is only accessible when a return value is available. ASP.NET Web Services throws System.InvalidOperationException when accessing the System.Web.Services.Protocols.SoapMessage.GetReturnValue property and the return value is not available.

SoapMessageStage enumeration (System.Web.Services.Protocols)
ToString

### Description

Specifies the processing stage of a SOAP message.

ASP.NET Web Services provides an extensibility mechanism for calling Web Services using SOAP. The extensibility mechanism revolves around a **System.Web.Services.Protocols.SoapExtension** that is allowed to inspect or modify a message at specific stages in message processing on either the client or the server. This enum specifies the processing stage of a

System.Web.Services.Protocols.SoapMessage .

**ToString** 

[C#] public const SoapMessageStage AfterDeserialize;

[C++] public: const SoapMessageStage AfterDeserialize;

[VB] Public Const AfterDeserialize As SoapMessageStage

[JScript] public var AfterDeserialize : SoapMessageStage; 1 2 Description 3 The stage just after a System. Web. Services. Protocols. Soap Message is 4 deserialized from a SOAP message into a object. 5 **ToString** 6 7 [C#] public const SoapMessageStage AfterSerialize; 8 [C++] public: const SoapMessageStage AfterSerialize; 9 [VB] Public Const AfterSerialize As SoapMessageStage 10 [JScript] public var AfterSerialize : SoapMessageStage; 11 12 Description 13 The stage just after a System. Web. Services. Protocols. Soap Message is 14 serialized, but before the SOAP message is sent over the wire. 15 **ToString** 16 17 [C#] public const SoapMessageStage BeforeDeserialize; 18 [C++] public: const SoapMessageStage BeforeDeserialize; 19 [VB] Public Const BeforeDeserialize As SoapMessageStage 20 [JScript] public var BeforeDeserialize : SoapMessageStage; 21 22 Description 23 The stage just before a System. Web. Services. Protocols. Soap Message is 24 deserialized from the SOAP message sent across the network into a object. 25

[C#] public const SoapMessageStage BeforeSerialize;

[C++] public: const SoapMessageStage BeforeSerialize;

[VB] Public Const BeforeSerialize As SoapMessageStage

[JScript] public var BeforeSerialize : SoapMessageStage;

### Description

The stage just prior to a **System.Web.Services.Protocols.SoapMessage** being serialized.

SoapParameterStyle enumeration (System.Web.Services.Protocols)
ToString

## Description

Specifies how parameters are encoded in the XML portion of a SOAP message.

System.Web.Services.Protocols.SoapParameterStyle can be set when applying a System.Web.Services.Protocols.SoapDocumentMethodAttribute or System.Web.Services.Protocols.SoapDocumentServiceAttribute attribute to a Web Service method or a method of a client proxy class and the class implementing a Web Service respectively.

**ToString** 

 $[C\#]\ public\ const\ Soap Parameter Style\ Bare;$ 

1	[C++] public: const SoapParameterStyle Bare;
2	[VB] Public Const Bare As SoapParameterStyle
3	[JScript] public var Bare : SoapParameterStyle;
4	
5	Description
6	Specifies the Web Service method parameters are elements directly beneath
7	the <b>Body</b> element of the XML portion of a SOAP request or SOAP response.
8	ToString
9	
10	[C#] public const SoapParameterStyle Default;
11	[C++] public: const SoapParameterStyle Default;
12	[VB] Public Const Default As SoapParameterStyle
13	[JScript] public var Default : SoapParameterStyle;
14	
15	Description
16	Specifies using the default SoapParameterStyle for the Web Service. The
17	default can for a Web Service by applying a SoapDocumentServiceAttribute to the
18	class implementing the Web Service. If a
19	System.Web.Services.Protocols.SoapDocumentServiceAttribute is not applied
20	to the class implementing the Web Service the default is
21	System.Web.Services.Protocols.SoapParameterStyle.Wrapped.
22	ToString
23	
24	[C#] public const SoapParameterStyle Wrapped;
25	[C++] public: const SoapParameterStyle Wrapped;

[VB] Public Const Wrapped As SoapParameterStyle [JScript] public var Wrapped : SoapParameterStyle;

Description

Specifies the Web Service method parameters are wrapped within a single element beneath the **Body** element of the XML portion of a SOAP request or SOAP response.

SoapRpcMethodAttribute class (System.Web.Services.Protocols)
ToString

Description

Applying the optional

System.Web.Services.Protocols.SoapRpcMethodAttribute to a Web Service method alters the format of the SOAP request or response sent to and from a Web Service method.

Web Services Description Language (WSDL) defines two styles for how a Web Service method, which it calls an operation, can be encoded in a SOAP request or a SOAP response: RPC and Document. The RPC style refers to encoding the Web Service method according to the SOAP specification for using SOAP for RPC; otherwise known as Section 7 of the SOAP specification. This style states that all parameters are wrapped within a single element named after the Web Service method and that each element within that element represent a parameter named after their respective parameter name.

Soap Rpc Method Attribute

1	Example Syntax:
2	ToString
3	
4	[C#] public SoapRpcMethodAttribute();
5	[C++] public: SoapRpcMethodAttribute();
6	[VB] Public Sub New()
7	[JScript] public function SoapRpcMethodAttribute(); Initializes a new instance of
8	the System. Web. Services. Protocols. Soap Rpc Method Attribute class.
9	
10	Description
11	Initializes a new instance of the
12	System.Web.Services.Protocols.SoapRpcMethodAttribute class setting all
13	properties to their defaults.
14	SoapRpcMethodAttribute
15	Example Syntax:
16	ToString
17	
18	[C#] public SoapRpcMethodAttribute(string action);
19	[C++] public: SoapRpcMethodAttribute(String* action);
20	[VB] Public Sub New(ByVal action As String)
21	[JScript] public function SoapRpcMethodAttribute(action : String);
22	
23	Description
24	Initializes a new instance of the
25	System.Web.Services.Protocols.SoapRpcMethodAttribute class setting the

1	System.Web.Services.Protocols.SoapRpcMethodAttribute.Action property to
2	action . The intent of the SOAP request. Sets the
3	$System. Web. Services. Protocols. Soap Rpc Method Attribute. Action \ property.$
4	Action
5	ToString
6	
7	[C#] public string Action {get; set;}
8	[C++] public:property String* get_Action();public:property void
9	set_Action(String*);
10	[VB] Public Property Action As String
11	[JScript] public function get Action(): String; public function set Action(String);
12	
13	Description
14	Gets or sets the intent of the SOAP request.
15	The System.Web.Services.Protocols.SoapRpcMethodAttribute.Action
16	property forms the SOAPAction HTTP Header Field for the HTTP Request.
17	Binding
18	ToString
19	
20	[C#] public string Binding {get; set;}
21	[C++] public:property String* get_Binding();public:property void
22	set_Binding(String*);
23	[VB] Public Property Binding As String
24	[JScript] public function get Binding(): String; public function set Binding(String);
25	

21

22

23

24

25

Description

1

3

4

Gets or sets the binding a Web Service method is implementing a operation for.

A binding, as defined by Web Services Description Language(WSDL), is similar to an interface, in that it defines a concrete set of operations. With respect to ASP.NET Web Services, each Web Service method is an operation within a binding. Web Service methods are members of either the default binding for a Web Service or in a binding specified within a

System. Web. Services. Web Service Binding Attribute applied to a Web Service.

A Web Service can implement multiple bindings, by applying multiple

System. Web. Services. Web Service Binding Attribute attributes to a Web Service.

OneWay

**ToString** 

[C#] public bool OneWay {get; set;}

[C++] public: \_\_property bool get\_OneWay();public: \_\_property void set OneWay(bool);

[VB] Public Property OneWay As Boolean

[JScript] public function get OneWay(): Boolean; public function set

OneWay(Boolean);

Description

Gets or sets whether a Web Service client waits for the Web server to finish processing a Web Service method.

1

3

5

6

7

8

9

16

17

18

19

20

21

22

23

24

25

When a Web Service method sets the

 $System. Web. Services. Protocols. Soap Rpc Method Attribute. One Way \ property$ to true, the Web Service client does not have to wait for the Web server to finish processing the Web Service method. As soon as the Web server has deserialized the System.Web.Services.Protocols.SoapServerMessage, but before invoking the Web Service method, the server returns an HTTP 202 status code. A HTTP 202 status code indicates to the client that the Web server has started processing of the message. Therefore, a Web Service client receives no acknowledgment that the Web server successfully processed the message.

RequestElementName

**ToString** 

[C#] public string RequestElementName {get; set;}

[C++] public: property String\* get RequestElementName();public: \_\_property void set RequestElementName(String\*);

[VB] Public Property RequestElementName As String

[JScript] public function get RequestElementName(): String;public function set RequestElementName(String);

Description

Gets or sets the XML element associated with the SOAP request for a Web Service method.

The

System. Web. Services. Protocols. Soap Rpc Method Attribute. Request Element Na

1	me defines the XML element used to wrap the parameters beneath the Body
2	element of the SOAP request.
3	RequestNamespace
4	ToString
5	
6	[C#] public string RequestNamespace {get; set;}
7	[C++] public:property String* get_RequestNamespace();public:property
8	void set_RequestNamespace(String*);
9	[VB] Public Property RequestNamespace As String
10	[JScript] public function get RequestNamespace() : String;public function set
11	RequestNamespace(String);
12	
13	Description
14	Gets or sets the XML namespace associated with the SOAP request for a
15	Web Service method.
16	ResponseElementName
17	ToString
18	
19	[C#] public string ResponseElementName {get; set;}
20	[C++] public:property String* get_ResponseElementName();public:property
21	void set_ResponseElementName(String*);
22	[VB] Public Property ResponseElementName As String
23	[JScript] public function get ResponseElementName(): String;public function set
24	ResponseElementName(String);
25	

18

19

20

21

22

23

24

25

Description

3

5

Gets or sets the XML element associated with the SOAP response for a Web Service method.

The

System.Web.Services.Protocols.SoapRpcMethodAttribute.ResponseElement
Name defines the XML element used to wrap the parameters beneath the Body
element of the SOAP response.

ResponseNamespace

**ToString** 

[C#] public string ResponseNamespace {get; set;}

[C++] public: \_\_property String\* get\_ResponseNamespace();public: \_\_property void set ResponseNamespace(String\*);

[VB] Public Property ResponseNamespace As String

[JScript] public function get ResponseNamespace() : String; public function set ResponseNamespace(String);

Description

Gets or sets the XML namespace associated with the SOAP response for a Web Service method.

The

System.Web.Services.Protocols.SoapRpcMethodAttribute.ResponseNamespa ce property is used in the XSD schema for the Web Service method in its Service Description.

TypeId SoapRpcServiceAttribute class (System.Web.Services.Protocols) 2 **ToString** 3 5 Description 6 Applying the optional 7 System.Web.Services.Protocols.SoapRpcServiceAttribute to a Web Service 8 sets the default format of SOAP requests and responses sent to and from Web 9 Service methods within the Web Service. 10 The System.Web.Services.Protocols.SoapRpcServiceAttribute allows 11 you to set the default encoding styles for Web Service methods within a Web 12 Service to use RPC method encoding and 13 System.Web.Services.Description.SoapBindingUse.Encoded parameter 14 encoding. 15 SoapRpcServiceAttribute 16 Example Syntax: 17 **ToString** 18 19 [C#] public SoapRpcServiceAttribute(); 20 [C++] public: SoapRpcServiceAttribute(); 21 [VB] Public Sub New() 22 [JScript] public function SoapRpcServiceAttribute(); 23 24 Description

1	Initializes a new instance of the
2	System.Web.Services.Protocols.SoapRpcServiceAttribute class.
3	RoutingStyle
4	ToString
5	
6	[C#] public SoapServiceRoutingStyle RoutingStyle {get; set;}
7	[C++] public:property SoapServiceRoutingStyle get_RoutingStyle();public:
8	property void set_RoutingStyle(SoapServiceRoutingStyle);
9	[VB] Public Property RoutingStyle As SoapServiceRoutingStyle
10	[JScript] public function get RoutingStyle(): SoapServiceRoutingStyle;public
11	function set RoutingStyle(SoapServiceRoutingStyle);
12	
13	Description
14	
15	TypeId
16	SoapServerMessage class (System.Web.Services.Protocols)
17	ToString
18	
19	
20	Description
21	Represents the data in a SOAP request received or a SOAP response sent
22	by a Web Service method at a specific
23	System.Web.Services.Protocols.SoapMessageStage .
24	Action
25	ToString

4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	

'	
2	[C#] public override string Action {get;}
3	[C++] public:property virtual String* get_Action();
4	[VB] Overrides Public ReadOnly Property Action As String
5	[JScript] public function get Action(): String;
6	
7	Description
8	Gets the SOAPAction HTTP request header field for the SOAP request or
9	SOAP response.
10	The System. Web. Services. Protocols. Soap Server Message. Action
11	property can be accessed during any
12	System.Web.Services.Protocols.SoapMessageStage .
13	ContentType
14	Exception
15	Headers
16	MethodInfo
17	ToString
18	
19	
20	Description
21	Gets a representation of the method prototype for the Web Service method
22	for which the SOAP request is intended.
23	Although the
24	System.Web.Services.Protocols.SoapServerMessage.MethodInfo property can
25	be accessed during any stage, the method information is only available during

System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize and  $System. Web. Services. Protocols. Soap Message Stage. Before Serialize \ .$ 2 OneWay 3 **ToString** 4 5 [C#] public override bool OneWay {get;} 6 [C++] public: \_\_property virtual bool get\_OneWay(); 7 [VB] Overrides Public ReadOnly Property OneWay As Boolean 8 [JScript] public function get OneWay(): Boolean; 9 10 Description 11 Gets a value indicating whether the client waits for the server to finish 12 processing a Web Service method. 13 A System. Web. Services. Protocols. Soap Document Method Attribute or 14 System.Web.Services.Protocols.SoapRpcMethodAttribute applied to a Web 15 Service method or Web Service client can specify whether the Web Service 16 method is one way or not by setting the 17 System. Web. Services. Protocols. Soap Document Method Attribute. One Wayne System. Web. Service and State of the State of Stat18 property of the attribute. 19 Server 20 **ToString** 21 22 [C#] public object Server {get;} 23 [C++] public: \_\_property Object\* get\_Server(); 24 [VB] Public ReadOnly Property Server As Object

[JScript] public function get Server(): Object;

Description

Gets the instance of the class handling the method invocation on the Web server System.Web.Services.Protocols.SoapMessageStage is not System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize or System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize .

The System.Web.Services.Protocols.SoapServerMessage.Server
property is an instance of the class implementing the Web Service. If a SOAP
extension knew ahead of time the type of the class, it could cast
System.Web.Services.Protocols.SoapServerMessage.Server to that type and
access properties and methods of the class implementing the Web Service.

Stage

Stream

Url

**ToString** 

Description

Gets the base url of the Web Service.

The System.Web.Services.Protocols.SoapServerMessage.Url property can be accessed during any System.Web.Services.Protocols.SoapMessageStage

EnsureInStage

1	
2	[C#] protected override void EnsureInStage();
3	[C++] protected: void EnsureInStage();
4	[VB] Overrides Protected Sub EnsureInStage()
5	[JScript] protected override function EnsureInStage();
6	
7	Description
8	Asserts that the current
9	System.Web.Services.Protocols.SoapMessageStage stage is a stage where in
10	parameters are available. If not, an exception is thrown.
11	For Web Service methods, the in parameters are available after the SOAP
12	request in the descrialization process in the
13	System.Web.Services.Protocols.SoapMessageStage.AfterDeserialize stage. The
14	System.InvalidOperationException is thrown if
15	System.Web.Services.Protocols.SoapServerMessage.EnsureInStage method is
16	invoked in any other System. Web. Services. Protocols. Soap Message Stage.
17	EnsureOutStage
18	
19	[C#] protected override void EnsureOutStage();
20	[C++] protected: void EnsureOutStage();
21	[VB] Overrides Protected Sub EnsureOutStage()
22	[JScript] protected override function EnsureOutStage();
23	
24	Description
25	

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Asserts that the current

**System.Web.Services.Protocols.SoapMessageStage** stage is a stage where out parameters are available. If not, an exception is thrown.

For Web Service methods, the out parameters are available prior to the SOAP response serialization process in the

System.Web.Services.Protocols.SoapMessageStage.BeforeSerialize stage. The System.InvalidOperationException is thrown if

 ${\bf System. Web. Services. Protocols. Soap Server Message. Ensure Out Stage \ method \\ is invoked in any other {\bf System. Web. Services. Protocols. Soap Message Stage} \ .$ 

SoapServiceRoutingStyle enumeration (System.Web.Services.Protocols)

**ToString** 

**ToString** 

**ToString** 

SoapUnknownHeader class (System.Web.Services.Protocols)

**ToString** 

# Description

Represents the data received from a SOAP header that was not understood by the recipient Web Service or Web Service client. This class cannot be inherited.

A SOAP client may invoke a Web Service with additional data beyond just the required parameters in the form of a SOAP header. An ASP.NET Web Service or Web Service client can view any SOAP headers it did not know about at the time the Web Service was written by applying a

System.Web.Services.Protocols.SoapHeaderAttribute with a

1	System. Web. Services. Protocols. Soap Header Attribute. Member Name property
2	of either an array of System. Web. Services. Protocols. Soap Header,
3	System.Web.Services.Protocols.SoapHeader,
4	System.Web.Services.Protocols.SoapUnknownHeader or an array of
5	System.Web.Services.Protocols.SoapUnknownHeader to the Web Service
6	method. Specifiying a System. Type of
7	System.Web.Services.Protocols.SoapUnknownHeader, allows the Web Service
8	to view the contents of the SOAP header and attempt to understand the semantics
9	of the SOAP header, in the form of an System.Xml.XmlElement.
10	SoapUnknownHeader
11	Example Syntax:
12	ToString
13	
14	[C#] public SoapUnknownHeader();
15	[C++] public: SoapUnknownHeader();
16	[VB] Public Sub New()
17	[JScript] public function SoapUnknownHeader();
18	Actor
19	DidUnderstand
20	Element
21	ToString
22	
23	
24	Description
25	Gets or sets the XML Header element for a SOAP request or response.

	1	If a Web Service method wants to process SOAP headers it doesn't know
	2	about at the time the Web Service is written, a Web Service method can process an
	3	System.Xml.XmlElement class representing the raw XML of the SOAP header.
	4	EncodedMustUnderstand
	5	MustUnderstand
	6	TextReturnReader class (System.Web.Services.Protocols)
	7	ToString
	8	
	9	
12 m	10	Description
	11	
The Man	12	TextReturnReader
	13	Example Syntax:
	14	ToString
	15	
	16	[C#] public TextReturnReader();
To the series	17	[C++] public: TextReturnReader();
	18	[VB] Public Sub New()
	19	[JScript] public function TextReturnReader();
	20	GetInitializer
	21	
	22	[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);
	23	[C++] public: Object* GetInitializer(LogicalMethodInfo* methodInfo);
	24	[VB] Overrides Public Function GetInitializer(ByVal methodInfo As
	25	LogicalMethodInfo) As Object

1	[JScript] public override function GetInitializer(methodInfo: LogicalMethodInfo)
2	: Object;
3	
4	Description
5	
6	Initialize
7	
8	[C#] public override void Initialize(object o);
9	[C++] public: void Initialize(Object* o);
10	[VB] Overrides Public Sub Initialize(ByVal o As Object)
11	[JScript] public override function Initialize(o : Object);
12	
13	Description
14	
15	Read
16	
17	[C#] public override object Read(WebResponse response, Stream
18	responseStream);
19	[C++] public: Object* Read(WebResponse* response, Stream* responseStream);
20	[VB] Overrides Public Function Read(ByVal response As WebResponse, ByVal
21	responseStream As Stream) As Object
22	[JScript] public override function Read(response : WebResponse, responseStream
23	: Stream): Object;
24	
25	Description

- 11	
1	
2	UrlEncodedParameterWriter class (System.Web.Services.Protocols)
3	ToString
4	
5	
6	Description
7	
8	UrlEncodedParameterWriter
9	Example Syntax:
10	ToString
11	
12	[C#] protected UrlEncodedParameterWriter();
13	[C++] protected: UrlEncodedParameterWriter();
14	[VB] Protected Sub New()
15	[JScript] protected function UrlEncodedParameterWriter();
16	RequestEncoding
17	ToString
18	
19	[C#] public override Encoding RequestEncoding {get; set;}
20	[C++] public:property virtual Encoding* get_RequestEncoding();public:
21	property virtual void set_RequestEncoding(Encoding*);
22	[VB] Overrides Public Property RequestEncoding As Encoding
23	[JScript] public function get RequestEncoding(): Encoding;public function set
24	RequestEncoding(Encoding);
25	

```
Description
2
3
           UsesWriteRequest
           Encode
5
6
    [C#] protected void Encode(TextWriter writer, object[] values);
    [C++] protected: void Encode(TextWriter* writer, Object* values gc[]);
8
    [VB] Protected Sub Encode(ByVal writer As TextWriter, ByVal values() As
    Object)
10
    [JScript] protected function Encode(writer : TextWriter, values : Object[]);
11
12
    Description
13
14
           Encode
15
16
    [C#] protected void Encode(TextWriter writer, string name, object value);
17
    [C++] protected: void Encode(TextWriter* writer, String* name, Object* value);
18
    [VB] Protected Sub Encode(ByVal writer As TextWriter, ByVal name As String,
19
    ByVal value As Object)
20
    [JScript] protected function Encode(writer: TextWriter, name: String, value:
21
    Object);
22
23
    Description
24
25
```

1	GetInitializer
2	
3	[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);
4	[C++] public: Object* GetInitializer(LogicalMethodInfo* methodInfo);
5	[VB] Overrides Public Function GetInitializer(ByVal methodInfo As
6	LogicalMethodInfo) As Object
7	[JScript] public override function GetInitializer(methodInfo: LogicalMethodInfo)
8	: Object;
9	
10	Description
11	
12	Initialize
13	
14	[C#] public override void Initialize(object initializer);
15	[C++] public: void Initialize(Object* initializer);
16	[VB] Overrides Public Sub Initialize(ByVal initializer As Object)
17	[JScript] public override function Initialize(initializer : Object);
18	
19	Description
20	
21	UrlParameterReader class (System.Web.Services.Protocols)
22	WriteRequest
23	
24	
25	Description

н	
1	
2	UrlParameterReader
3	Example Syntax:
4	WriteRequest
5	
6	[C#] public UrlParameterReader();
7	[C++] public: UrlParameterReader();
8	[VB] Public Sub New()
9	[JScript] public function UrlParameterReader();
10	Read
11	
12	[C#] public override object[] Read(HttpRequest request);
13	[C++] public: Object* Read(HttpRequest* request)gc[];
14	[VB] Overrides Public Function Read(ByVal request As HttpRequest) As Object()
15	[JScript] public override function Read(request : HttpRequest) : Object[];
16	
17	Description
18	
19	UrlParameterWriter class (System.Web.Services.Protocols)
20	ToString
21	
22	
23	Description
24	
25	UrlParameterWriter
	···

```
Example Syntax:
           ToString
2
3
    [C#] public UrlParameterWriter();
4
    [C++] public: UrlParameterWriter();
5
    [VB] Public Sub New()
6
    [JScript] public function UrlParameterWriter();
           RequestEncoding
8
           UsesWriteRequest
9
           GetRequestUrl
10
11
    [C#] public override string GetRequestUrl(string url, object[] parameters);
12
    [C++] public: String* GetRequestUrl(String* url, Object* parameters __gc[]);
13
    [VB] Overrides Public Function GetRequestUrl(ByVal url As String, ByVal
14
    parameters() As Object) As String
15
    [JScript] public override function GetRequestUrl(url : String, parameters :
16
    Object[]): String;
17
18
    Description
19
20
           ValueCollectionParameterReader class (System.Web.Services.Protocols)
21
           WriteRequest
22
23
24
    Description
```

	ll.	
	1	
	2	ValueCollectionParameterReader
	3	Example Syntax:
	4	WriteRequest
	5	
	6	[C#] protected ValueCollectionParameterReader();
	7	[C++] protected: ValueCollectionParameterReader();
	8	[VB] Protected Sub New()
	9	[JScript] protected function ValueCollectionParameterReader();
i i	10	GetInitializer
	11	
	12	[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);
	13	[C++] public: Object* GetInitializer(LogicalMethodInfo* methodInfo);
	14	[VB] Overrides Public Function GetInitializer(ByVal methodInfo As
	15	LogicalMethodInfo) As Object
	16	[JScript] public override function GetInitializer(methodInfo: LogicalMethodInfo)
	17	: Object;
	18	
	19	Description
	20	
	21	Initialize
	22	
	23	[C#] public override void Initialize(object o);
	24	[C++] public: void Initialize(Object* o);
	25	[VB] Overrides Public Sub Initialize(ByVal o As Object)
		II

[JScript] public override function Initialize(o : Object);
Description
IsSupported
[C#] public static bool IsSupported(LogicalMethodInfo methodInfo);
[C++] public: static bool IsSupported(LogicalMethodInfo* methodInfo);
[VB] Public Shared Function IsSupported(ByVal methodInfo As
LogicalMethodInfo) As Boolean
[JScript] public static function IsSupported(methodInfo : LogicalMethodInfo) :
Boolean;
Description
IsSupported
[C#] public static bool IsSupported(ParameterInfo paramInfo);
[C++] public: static bool IsSupported(ParameterInfo* paramInfo);
[VB] Public Shared Function IsSupported(ByVal paramInfo As ParameterInfo) As
Boolean
[JScript] public static function IsSupported(paramInfo : ParameterInfo) : Boolean;
Description

23

24

 $[C++]\ protected:\ Object*\ Read(NameValueCollection*\ collection)\ \_\_gc[];$ 4 [VB] Protected Function Read(ByVal collection As NameValueCollection) As 5 Object() 6  $[JScript]\ protected\ function\ Read(collection: NameValueCollection): Object[];$ 8 Description 9 10 11 **ToString** 12 13 14 Description15 16 AsyncState 17 **ToString** 18 19 [C#] public object AsyncState {get;} 20 [C++] public: \_\_property Object\* get\_AsyncState(); 21

[C#] protected object[] Read(NameValueCollection collection);

Read

2

3

WebClientAsyncResult class (System.Web.Services.Protocols) [VB] Public ReadOnly Property AsyncState As Object [JScript] public function get AsyncState(): Object; Description 952

	1	
	2	AsyncWaitHandle
	3	ToString
	4	
	5	[C#] public WaitHandle AsyncWaitHandle {get;}
	6	[C++] public:property WaitHandle* get_AsyncWaitHandle();
	7	[VB] Public ReadOnly Property AsyncWaitHandle As WaitHandle
	8	[JScript] public function get AsyncWaitHandle(): WaitHandle;
	9	
	10	Description
նում մում նութ մուս վուն ձուն նույն կում	11	
ili init	12	CompletedSynchronously
in Tank Thank	13	ToString
	14	
	15	[C#] public bool CompletedSynchronously {get;}
Fluid Start	16	[C++] public:property bool get_CompletedSynchronously();
ii S	17	[VB] Public ReadOnly Property CompletedSynchronously As Boolean
	18	[JScript] public function get CompletedSynchronously(): Boolean;
	19	
	20	Description
	21	Gets a value indicating whether the invocation of the Web Service method
	22	completed synchronously.
	23	IsCompleted
	24	ToString
	25	

1	
2	[C#] public bool IsCompleted {get;}
3	[C++] public:property bool get_IsCompleted();
4	[VB] Public ReadOnly Property IsCompleted As Boolean
5	[JScript] public function get IsCompleted(): Boolean;
6	
7	Description
8	Gets a value indicating whether the asynchronous request has completed.
9	Abort
10	
11	[C#] public void Abort();
12	[C++] public: void Abort();
13	[VB] Public Sub Abort()
14	[JScript] public function Abort();
15	
16	Description
17	
18	WebClientProtocol class (System.Web.Services.Protocols)
19	ToString
20	
21	
22	Description
23	Specifies the base class for all ASP.NET Web Service client proxies.
24	ASP.NET incorporates two distinct functionalities of Web Services:
25	building ASP.NET Web Services and building Web Service clients. If you are

	1	building a Web Service client using ASP.NET, you must create a proxy class
	2	deriving indirectly or directly from
	3	System.Web.Services.Protocols.WebClientProtocol for the Web Service you
	4	want to call.
	5	WebClientProtocol
	6	Example Syntax:
	7	ToString
	8	
	9	[C#] protected WebClientProtocol();
,	10	[C++] protected: WebClientProtocol();
in the state of	11	[VB] Protected Sub New()
	12	[JScript] protected function WebClientProtocol();
նում նոմ կոմ հոռ հոռ հոմ նոմ կոմ կոմ	13	
	14	Description
II. iiiiii dad	15	Initializes a new instance of the
in in	16	System.Web.Services.Protocols.WebClientProtocol class.
=	17	ConnectionGroupName
	18	ToString
	19	
	20	[C#] public string ConnectionGroupName {get; set;}
	21	[C++] public:property String* get_ConnectionGroupName();public:property
	22	<pre>void set_ConnectionGroupName(String*);</pre>
	23	[VB] Public Property ConnectionGroupName As String
	24	[JScript] public function get ConnectionGroupName(): String; public function set
	25	ConnectionGroupName(String);

#### Description

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the name of the connection group for the request.

The

System. Web. Services. Protocols. Web Client Protocol. Connection Group Nameproperty enables you to associate a request to a connection group. For more  $details, see {\bf \, System. Net. Http WebRequest. Connection Group Name}\;.$ 

Container

Credentials

**ToString** 

### Description

Gets or sets security credentials for Web Service client authentication.

When using the Credentials property, a Web Service client must instantiate a class implementing System.Net.ICredentials, such as System.Net.NetworkCredential, and then set the client credentials specific to the authentication mechanism. The System.Net.NetworkCredential class can be used to set authentication credentials using the basic, digest, NTLM and Kerberos authentication mechanisms.

956

DesignMode

**Events** 

PreAuthenticate

**ToString** 

lee@hayes plic 509-324-9256

2	
3	Description
4	Gets or sets whether pre-authentication is enabled.
5	When
6	System.Web.Services.Protocols.WebClientProtocol.PreAuthenticate is true
7	the WWW-authenticate header is sent with each request back to the server,
8	otherwise standard authentication procedures are used. When
9	System.Web.Services.Protocols.WebClientProtocol.PreAuthenticate is false
10	the authentication headers are sent only on the first request.
11	RequestEncoding
12	ToString
13	
14	[C#] public Encoding RequestEncoding {get; set;}
15	[C++] public:property Encoding* get_RequestEncoding();public:property
16	void set_RequestEncoding(Encoding*);
17	[VB] Public Property RequestEncoding As Encoding
18	[JScript] public function get RequestEncoding(): Encoding; public function set
19	RequestEncoding(Encoding);
20	
21	Description
22	The <b>System.Text.Encoding</b> used to make the client request to the Web
23	Service.
24	The
25	System.Web.Services.Protocols.WebClientProtocol.RequestEncoding

```
determines the encoding for the request message. The
1
    System.Net.WebRequest.ContentType of the request will be annotated with the
2
    encoding value.
3
           Site
           Timeout
5
           ToString
8
    Description
9
           Indicates the time a Web Service client waits for a synchronous Web
10
    Service request to complete (in milliseconds).
11
           Even though a Web Service client can set the
12
    System.Web.Services.Protocols.WebClientProtocol.Timeout property to
13
    infinity with a value of -1, the Web Server can still cause the request to time out
14
    on the server side.
15
           Url
16
           ToString
17
18
    [C#] public string Url {get; set;}
19
    [C++] public: __property String* get_Url();public: __property void
20
    set Url(String*);
21
    [VB] Public Property Url As String
22
    [JScript] public function get Url(): String; public function set Url(String);
23
24
    Description
```

24

Gets or sets the base URL of the Web Service the client is requesting.

Proxy classes generated using Wsdl.exe will set a default

System.Web.Services.Protocols.WebClientProtocol.Url property for the client to use. The default System.Web.Services.Protocols.WebClientProtocol.Url is determined by the location attribute found in the Service Description from which the proxy class was generated.

Abort

[C#] public virtual void Abort();

[C++] public: virtual void Abort();

[VB] Overridable Public Sub Abort()

[JScript] public function Abort();

Description

Cancels a request to a Internet resource.

Return Value: System.Web.Services.Protocols.WebClientProtocol.Abort cancels a request to a resource made using the

Uri) method. After the request is canceled, invoking

System. Web. Services. Protocols. Web Client Protocol. Get Web Response (System. Response) and the protocol of the protocol

Net.WebRequest) will cause a System.Net.WebException .

AddToCache

[C#] protected static void AddToCache(Type type, object value);

[C++] protected: static void AddToCache(Type\* type, Object\* value);

1	[VB] Protected Shared Sub AddToCache(ByVal type As Type, ByVal value As
2	Object)
3	[JScript] protected static function AddToCache(type: Type, value: Object);
4	
5	Description
6	Add an instance of the client protocol handler to the cache.
7	GetFromCache
8	
9 [	[C#] protected static object GetFromCache(Type type);
10	[C++] protected: static Object* GetFromCache(Type* type);
11	[VB] Protected Shared Function GetFromCache(ByVal type As Type) As Object
12	[JScript] protected static function GetFromCache(type: Type): Object;
13	
14	Description
15	Gets an instance of a client protocol handler from the cache.
16	GetWebRequest
17	
18	[C#] protected virtual WebRequest GetWebRequest(Uri uri);
19	[C++] protected: virtual WebRequest* GetWebRequest(Uri* uri);
20	[VB] Overridable Protected Function GetWebRequest(ByVal uri As Uri) As
21	WebRequest
22	[JScript] protected function GetWebRequest(uri : Uri) : WebRequest;
23	
24	Description
25	

3

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

25

Creates a **System.Net.WebRequest** instance for the specified *url* .

Return Value: The **System.Net.WebRequest** instance.

This method can be overridden to customize the **System.Net.WebRequest** object before the Web Service request is made. For example you could add a custom header to the request. The **System.Uri** to use when creating the **System.Net.WebRequest**.

GetWebResponse

[C#] protected virtual WebResponse GetWebResponse(WebRequest request);

[C++] protected: virtual WebResponse\* GetWebResponse(WebRequest\* request);

[VB] Overridable Protected Function GetWebResponse(ByVal request As

WebRequest) As WebResponse

[JScript] protected function GetWebResponse(request : WebRequest) :

WebResponse; Returns a response from a request to a Web Service method.

#### Description

Returns a response from a synchronous request to a Web Service method.

Return Value: A System.Net.WebResponse. The System.Net.WebRequest to get the response from.

GetWebResponse

IAsyncResult\* result);

[C#] protected virtual WebResponse GetWebResponse(WebRequest request,IAsyncResult result);[C++] protected: virtual WebResponse\* GetWebResponse(WebRequest\* request,

lee@hayes.pik 509-324-9256 961 *MSI-863US.APP* 

1	[VB] Overridable Protected Function GetWebResponse(ByVal request As
2	WebRequest, ByVal result As IAsyncResult) As WebResponse
3	[JScript] protected function GetWebResponse(request : WebRequest, result :
4	IAsyncResult): WebResponse;
5	
6	Description
7	Returns a response from an asynchronous request to a Web Service method.
8	Return Value: The System.Net.WebResponse instance. The
9	System.Net.WebRequest to get the response from. The System.IAsyncResult to
10	pass to System.Net.HttpWebRequest.EndGetResponse(System.IAsyncResult).
11	when the response has completed.
12	WebServiceHandlerFactory class (System.Web.Services.Protocols)
13	ToString
14	
15	
16	Description
17	
18	WebServiceHandlerFactory
19	Example Syntax:
20	ToString
21	
22	[C#] public WebServiceHandlerFactory();
23	[C++] public: WebServiceHandlerFactory();
24	[VB] Public Sub New()
25	[JScript] public function WebServiceHandlerFactory();

GetHandler

2

3

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] public IHttpHandler GetHandler(HttpContext context, string verb, string url, string filePath);

[C++] public: \_\_sealed IHttpHandler\* GetHandler(HttpContext\* context, String\* verb, String\* url, String\* filePath);

[VB] NotOverridable Public Function GetHandler(ByVal context As HttpContext, ByVal verb As String, ByVal url As String, ByVal filePath As String) As

IHttpHandler

[JScript] public function GetHandler(context : HttpContext, verb : String, url :

String, filePath: String): IHttpHandler;

Description

# ReleaseHandler

[C#] public void ReleaseHandler(IHttpHandler handler);

[C++] public: \_\_sealed void ReleaseHandler(IHttpHandler\* handler);

[VB] NotOverridable Public Sub ReleaseHandler(ByVal handler As

IHttpHandler)

[JScript] public function ReleaseHandler(handler: IHttpHandler);

Description

XmlReturnReader class (System.Web.Services.Protocols)

1	ToString
2	
3	
4	Description
5	
6	XmlReturnReader
7	Example Syntax:
8	ToString
9	
10	[C#] public XmlReturnReader();
11	[C++] public: XmlReturnReader();
12	[VB] Public Sub New()
13	[JScript] public function XmlReturnReader();
14	GetInitializer
15	
16	[C#] public override object GetInitializer(LogicalMethodInfo methodInfo);
17	[C++] public: Object* GetInitializer(LogicalMethodInfo* methodInfo);
18	[VB] Overrides Public Function GetInitializer(ByVal methodInfo As
19	LogicalMethodInfo) As Object
20	[JScript] public override function GetInitializer(methodInfo: LogicalMethodInfo)
21	: Object;
22	
23	Description
24	
25	GetInitializers

1	
2	[C#] public override object[] GetInitializers(LogicalMethodInfo[] methodInfos);
3	[C++] public: Object* GetInitializers(LogicalMethodInfo* methodInfos[])gc[]
4	[VB] Overrides Public Function GetInitializers(ByVal methodInfos() As
5	LogicalMethodInfo) As Object()
6	[JScript] public override function GetInitializers(methodInfos :
7	LogicalMethodInfo[]) : Object[];
8	
9	Description
10	
11	Initialize
12	
13	[C#] public override void Initialize(object o);
14	[C++] public: void Initialize(Object* o);
15	[VB] Overrides Public Sub Initialize(ByVal o As Object)
16	[JScript] public override function Initialize(o : Object);
17	
18	Description
19	
20	Read
21	
22	[C#] public override object Read(WebResponse response, Stream
23	responseStream);
24	
25	

1	
2	System.Web.SessionState
3	
4	Description
5	
6	HttpSessionState class (System.Web.SessionState)
7	
8	
9	Description
10	Provides access to session state values as well as session-level settings and
11	lifetime management methods.
12	Properties:
13	CodePage
14	
15	[C#] public int CodePage {get; set;}
16	[C++] public:property int get_CodePage();public:property void
17	set_CodePage(int);
18	[VB] Public Property CodePage As Integer
19	[JScript] public function get CodePage(): int;public function set CodePage(int);
20	
21	Description
22	Gets or sets the code page identifier for the current session.
23	Contents
24	
25	[C#] public HttpSessionState Contents {get;}

1	[C++] public:property HttpSessionState* get_Contents();
2	[VB] Public ReadOnly Property Contents As HttpSessionState
3	[JScript] public function get Contents(): HttpSessionState;
4	
5	Description
6	Gets a reference to the current session state object.
7	Contents is provided for legacy ASP compatibility.
8	Count
9	
10	[C#] public int Count {get;}
11	[C++] public:property int get_Count();
12	[VB] Public ReadOnly Property Count As Integer
13	[JScript] public function get Count(): int;
14	
15	Description
16	Gets the number of items in the session state collection
17	IsCookieless
18	
19	[C#] public bool IsCookieless {get;}
20	[C++] public:property bool get_IsCookieless();
21	[VB] Public ReadOnly Property IsCookieless As Boolean
22	[JScript] public function get IsCookieless(): Boolean;
23	
24	Description
25	

1	Gets a value indicating whether the session is managed using cookieless
2	session.
3	IsNewSession
4	
5	[C#] public bool IsNewSession {get;}
6	[C++] public:property bool get_IsNewSession();
7	[VB] Public ReadOnly Property IsNewSession As Boolean
8	[JScript] public function get IsNewSession(): Boolean;
9	
10	Description
11	Gets a value indicating whether the session has been created with the
12	current request.
13	IsReadOnly
14	
15	[C#] public bool IsReadOnly {get;}
16	[C++] public:property bool get_IsReadOnly();
17	[VB] Public ReadOnly Property IsReadOnly As Boolean
18	[JScript] public function get IsReadOnly(): Boolean;
19	
20	Description
21	Gets a value indicating whether the session is read-only.
22	IsSynchronized
23	
24	[C#] public bool IsSynchronized {get;}
25	[C++] public:property bool get_IsSynchronized();

	1	[VB] Public ReadOnly Property IsSynchronized As Boolean
	2	[JScript] public function get IsSynchronized(): Boolean;
	3	
	4	Description
	5	Gets a value indicating whether access to the collection of session state
	6	values is read-only (thread-safe).
	7	Item
	8	
	9	[C#] public object this[string name] {get; set;}
4.4	10	[C++] public:property Object* get_Item(String* name);public:property void
and their that their	11	set_Item(String* name, Object*);
	12	[VB] Public Default Property Item(ByVal name As String) As Object
	13	[JScript] returnValue =
	14	HttpSessionStateObject.Item(name);HttpSessionStateObject.Item(name) =
	15	returnValue; Gets or sets individual session values.
	16	
	17	Description
	18	Gets or sets a session value by name. The key name of the session value.
	19	Item
	20	
	21	[C#] public object this[int index] {get; set;}
	22	[C++] public:property Object* get_Item(int index);public:property void
	23	set_Item(int index, Object*);
	24	[VB] Public Default Property Item(ByVal index As Integer) As Object
	25	[JScript] returnValue =

1	HttpSessionStateObject.Item(index);HttpSessionStateObject.Item(index) =
2	returnValue;
3	
4	Description
5	Gets or sets a session value by numerical index. The numerical index of the
6	session value.
7	Keys
8	
9	[C#] public NameObjectCollectionBase.KeysCollection Keys {get;}
10	[C++] public:property NameObjectCollectionBase.KeysCollection*
11	get_Keys();
12	[VB] Public ReadOnly Property Keys As
13	NameObjectCollectionBase.KeysCollection
14	[JScript] public function get Keys(): NameObjectCollectionBase.KeysCollection;
15	
16	Description
17	Gets a collection of all session keys.
18	LCID
19	
20	[C#] public int LCID {get; set;}
21	[C++] public:property int get_LCID();public:property void set_LCID(int);
22	[VB] Public Property LCID As Integer
23	[JScript] public function get LCID(): int;public function set LCID(int);
24	
25	Description

1	Gets or sets the locale identifier (LCID) of the current session.
2	Mode
3	
4	[C#] public SessionStateMode Mode {get;}
5	[C++] public:property SessionStateMode get_Mode();
6	[VB] Public ReadOnly Property Mode As SessionStateMode
7	[JScript] public function get Mode() : SessionStateMode;
8	
9	Description
10	Gets the current session state mode.
11	SessionID
12	
13	[C#] public string SessionID {get;}
14	[C++] public:property String* get_SessionID();
15	[VB] Public ReadOnly Property SessionID As String
16	[JScript] public function get SessionID(): String;
17	
18	Description
19	Gets the unique session ID used to identify a session.
20	StaticObjects
21	
22	[C#] public HttpStaticObjectsCollection StaticObjects {get;}
23	[C++] public:property HttpStaticObjectsCollection* get_StaticObjects();
24	[VB] Public ReadOnly Property StaticObjects As HttpStaticObjectsCollection
25	[JScript] public function get StaticObjects(): HttpStaticObjectsCollection;

Description Gets

Gets a collection of objects declared by

#### System.Web.UI

#### Description

The System.Web.UI namespace provides classes and interfaces that allow you to create controls and pages that will appear in your Web applications as user interface on a Web page. This namespace includes the Control class, which provides all controls, whether HTML, Web, or User controls, with a common set of functionality. It also includes the Page control, which is generated automatically whenever a request is made for a page in your Web application. Also provided are classes which provide the Web Forms Server Controls data binding functionality, the ability to save the view state of a given control or page, as well as parsing functionality for both programmable and literal controls.

AttributeCollection class (System.Web.UI)

### Description

Provides object-model access to all attributes declared in the opening tag of an ASP.NET server control element. This class cannot be inherited.

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Attributes are case-insensitive strings. They return a **System.String** object as their value. If there are no attributes in the collection, they return **null**.

Constructors:

AttributeCollection

Example Syntax:

AttributeCollection(StateBag public bag); [C#] public: AttributeCollection(StateBag\* bag); [C++][VB] Public Sub New(ByVal bag As StateBag) public function AttributeCollection(bag StateBag); [JScript]

Description

Initializes a new instance of the **System.Web.UI.AttributeCollection** class. A **System.Web.UI.StateBag** object that contains the attribute keys and their values that are in the opening tag of the server control.

Properties:

Count

[C#] public int Count {get;} public: get Count(); [C++]\_ property int ReadOnly [VB] **Public** Property Count As Integer function Count() [JScript] public int; get :

Description

Gets the number of attributes in the **System.Web.UI.AttributeCollection** object.

CssStyle

1

2

3

5

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] public CssStyleCollection CssStyle {get;} get CssStyle(); public: CssStyleCollection\* [C++]property [VB] Public ReadOnly Property CssStyle CssStyleCollection As [JScript] public function CssStyle() CssStyleCollection; get

### Description

Gets a collection of styles for the ASP.NET server control to which the current System.Web.UI.AttributeCollection object belongs.

Any style declared for a particular HTML server control is added to the **CssStyleCollection** object when the containing Web Forms page is parsed. You can use this property to add, remove and iterate through the styles declared for the server control.

Item

[C#] public string this[string key] {get; set;} [C++] public: property String\* get\_Item(String\* key);public: \_\_property void set Item(String\* String\*); key, [VB] Public Default Property Item(ByVal key As String) As [JScript] returnValue AttributeCollectionObject.Item(key); AttributeCollectionObject.Item(key) returnValue;

# Description

1

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

20

21

22

23

24

25

Gets or sets a specified attribute value for a server control. The location of the attribute within the collection.

Keys

[C#]	publ	ic I	ICollection		Keys	
[C++]	public:	proj	perty	ICollection	n*	get_Keys();
[VB]	Public	ReadOnly	Property	Keys	As	ICollection
[JScript]	public	function	get	Keys()	:	ICollection;

# Description

Gets a collection of keys to all attributes in the server control's **System.Web.UI.AttributeCollection** object.

Methods:

Add

```
public
                                Add(string
[C#]
                     void
                                               key,
                                                         string
                                                                    value);
[C++]
          public:
                               Add(String*
                      void
                                                key,
                                                         String*
                                                                    value);
[VB] Public Sub Add(ByVal key As String, ByVal value As
                                                                    String)
[JScript]
           public
                    function
                               Add(key
                                              String,
                                                        value
                                                                    String);
```

Description

lee@hayes plic 509-324-9256

Adds an attribute to a server contol's **System.Web.UI.AttributeCollection** object. The index assigned to the new attribute in the collection. The attribute to store in the collection.

#### AddAttributes

[C#]	public	void	AddAttribu	tes(Html7	[extW	riter	writer);
[C++]	public:	void	AddAttribut	es(HtmlT	'extW	riter*	writer);
[VB]	Public Sub	AddAtt	ributes(ByVal	writer	As	HtmlTe	xtWriter)
[JScrip	t] public	function	AddAttribute	s(writer	:	HtmlTe	xtWriter);

#### Description

Adds attributes from the **AttributeCollection** class to the **System.Web.UI.HtmlTextWriter** object that is responsible for rendering the attributes as HTML to an ASP.NET server control.

This method copies all of the server control's attributes to an System.Web.UI.HtmlTextWriter object so they can be rendered by the next call to the System.Web.UI.HtmlTextWriter.RenderBeginTag(System.String) method. An System.Web.UI.HtmlTextWriter that writes the added attribute to the opening tag of an ASP.NET server control.

#### Clear

[C#]	public	void	Clear();
[C++]	public:	void	Clear();
[VB]	Public	Sub	Clear()
[JScript]	public	function	Clear();

1 Description 2 all attributes from control's Removes a server 3 System.Web.UI.AttributeCollection object. Remove 5 6 public Remove(string [C#] void key); 7 [C++]public: void Remove(String\* key); 8 [VB] Public Sub Remove(ByVal key As String) ġ function [JScript] public Remove(key String); 10 11 Description 12 Removes attribute from control's server an a 13 System. Web. UI. Attribute Collection object. The index of the attribute to remove. 14 Render 15 16 public Render(HtmlTextWriter [C#] void writer); 17 [C++]public: void Render(HtmlTextWriter\* writer); 18 [VB] Public Render(ByVal Sub writer As HtmlTextWriter) 19 function [JScript] public Render(writer HtmlTextWriter); : 20 21 Description 22 Writes the collection of attributes the specified to 23 System.Web.UI.HtmlTextWriter output stream. In turn, the output stream writes 24 25

the collection to the Web Forms page. An System.Web.UI.HtmlTextWriter object that writes the attribute collection to the opening HTML tag. AutomaticHandlerMethodInfos class (System.Web.UI) 3 **ToString** 5 6 Description AutomaticHandlerMethodInfos 8 Example Syntax: 9 **ToString** 10 11 AutomaticHandlerMethodInfos(); public [C#] 12 public: AutomaticHandlerMethodInfos(); [C++]13 [VB] Public Sub New() 14 [JScript] public function AutomaticHandlerMethodInfos(); 15 BaseParser class (System.Web.UI) 16 **ToString** 17 18 19 Description 20 Provides a base set of functionality for all parsers involved in parsing 21 ASP.NET page requests. This includes parsing of all ASP.NET server controls, 22 including pages and user controls. 23 BaseParser 24

Example Syntax:

```
ToString
    [C#]
                                                                        BaseParser();
                                     public
    [C++]
                                     public:
                                                                        BaseParser();
                                                                               New()
    [VB]
                             Public
                                                       Sub
    [JScript] public function BaseParser();
           BasePartialCachingControl class (System.Web.UI)
7
           ToString
8
9
10
    Description
11
                                                        functionality
           Provides
                                                 of
                                                                          for
                                                                                  the
                               base
                        a
                                         set
12
    System.Web.UI.StaticPartialCachingControl
                                                                                  the
                                                                 and
13
    System.Web.UI.PartialCachingControl classes.
14
           BasePartialCachingControl
15
           Example Syntax:
16
           ToString
17
18
                                                        BasePartialCachingControl();
    [C#]
                            protected
19
                            protected:
                                                        BasePartialCachingControl();
    [C++]
20
    [VB]
                            Protected
                                                        Sub
                                                                               New()
21
    [JScript] protected function BasePartialCachingControl();
22
           ChildControlsCreated
23
           ClientID
24
           Context
25
```

1	Controls			
2	Dependency			
3	ToString			
4				
5				
6	Description			
7	Gets or sets an instan	ce of the System.Wo	eb.Caching.Cacl	neDependency
8	class associated with the cach	ed server control outp	ut.	
9	EnableViewState			
10	Events			
11	HasChildViewState			
12	ID			
13	IsTrackingViewState			
14	NamingContainer			
15	Page			
16	Parent			
17	Site			
18	TemplateSourceDirec	tory		
19	UniqueID			
20	ViewState			
21	ViewStateIgnoresCase	e		
22	Visible			
23	Dispose			
24				
25	[C#] public	override	void	Dispose();

[C++] public: void Dispose();
[VB] Overrides Public Sub Dispose()
[JScript] public override function Dispose();

Description

OnInit

 $_{1}\parallel$ 

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] protected override void OnInit(EventArgs e); [C++] protected: OnInit(EventArgs\* void e); [VB] Overrides OnInit(ByVal EventArgs) Protected Sub As function [JScript] protected override OnInit(e EventArgs);

# Description

Raises the Init event for the server control to be output cached.

This method checks the cache for the content associated with the server control. If it is not there, the server control is created and added to the parent control's **System.Web.UI.ControlCollection** object. An **System.EventArgs** object that contains the event data.

Render

[C#] protected override void Render(HtmlTextWriter output);
[C++] protected: void Render(HtmlTextWriter\* output);
[VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)
[JScript] protected override function Render(output : HtmlTextWriter);

1	
2	Description
3	Outputs the server control's content to the
4	System.Web.UI.HtmlTextWriter output stream. The
5	System.Web.UI.HtmlTextWriter object that writes the cached control to the
6	page.
7	BuildMethod delegate (System.Web.UI)
8	TrackViewState
9	
10	
11	Description
12	Represents the method that is used to build a System.Web.UI.Control.
13	This delegate passes the information necessary to build ASP.NET server
14	controls as defined by the TemplateControl class, or one of the classes that inherit
15	from TemplateControl, Page and UserControl.
16	BuildTemplateMethod delegate (System.Web.UI)
17	TrackViewState
18	
19	
20	Description
21	
22	CompiledTemplateBuilder class (System.Web.UI)
23	TrackViewState
24	
25	

```
2
    Description
          CompiledTemplateBuilder
5
          Example Syntax:
6
          TrackViewState
8
                                  CompiledTemplateBuilder(BuildTemplateMethod
    [C#]
                  public
    buildTemplateMethod);
10
                                 CompiledTemplateBuilder(BuildTemplateMethod*
    [C++]
                  public:
11
    buildTemplateMethod);
12
    [VB] Public Sub New(ByVal buildTemplateMethod As BuildTemplateMethod)
13
    [JScript] public function CompiledTemplateBuilder(buildTemplateMethod :
14
    BuildTemplateMethod);
15
16
    Description
17
18
          InstantiateIn
19
20
    [C#]
                public
                               void
                                           InstantiateIn(Control
                                                                       container);
21
    [C++]
              public:
                           sealed
                                      void
                                              InstantiateIn(Control*
                                                                        container);
22
    [VB] NotOverridable Public Sub InstantiateIn(ByVal container As Control)
23
                                         InstantiateIn(container
    [JScript]
                 public
                            function
                                                                         Control);
24
25
```

11	
1	
2	Description
3	
4	ConstructorNeedsTagAttribute class (System.Web.UI)
5	ToString
6	
7	
8	Description
9	Specifies that a server control needs a tag name in its constructor.
10	ConstructorNeedsTagAttribute
11	Example Syntax:
12	ToString
13	
14	[C#] public ConstructorNeedsTagAttribute();
15	[C++] public: ConstructorNeedsTagAttribute();
16	[VB] Public Sub New()
17	[JScript] public function ConstructorNeedsTagAttribute(); Initializes a new
18	instance of the System. Web.UI. Constructor Needs Tag Attribute class.
19	
20	Description
21	Initializes a new instance of the
22	System.Web.UI.ConstructorNeedsTagAttribute class.
23	ConstructorNeedsTagAttribute
24	Example Syntax:
25	ToString

public ConstructorNeedsTagAttribute(bool [C#] needsTag); 2 [C++]public: ConstructorNeedsTagAttribute(bool needsTag); [VB] Public Sub New(ByVal needsTag As Boolean) [JScript] public function ConstructorNeedsTagAttribute(needsTag : Boolean); 6 Description 7 **Initializes** instance of the a new 8 System. Web. UI. Constructor Needs Tag Attribute class. true to add a tag to a 9 control; otherwise, false. 10 NeedsTag 11 **ToString** 12 13 public NeedsTag [C#] bool {get;} 14 public: bool get NeedsTag(); [C++]property 15 Public ReadOnly [VB] Property NeedsTag As Boolean 16 [JScript] public function NeedsTag() Boolean; get 17 18 Description 19 Indicates whether a control needs a tag in its contstructor. This property is 20 read-only. 21 **TypeId** 22 Control class (System.Web.UI) 23 **ToString** 24 25

Description

Defines the properties, methods, and events that are shared by all ASP.NET server controls.

This is the primary class that you derive from when you develop custom ASP.NET server controls.

Control

Example Syntax:

**ToString** 

[C#] public Control();

[C++] public: Control();

[VB] Public Sub New()

[JScript] public function Control();

Description

Initializes a new instance of the **System.Web.UI.Control** class.

ChildControlsCreated

**ToString** 

[C#] protected bool ChildControlsCreated {get; set;}

[C++] protected: \_\_property bool get\_ChildControlsCreated();protected:

\_\_property void set\_ChildControlsCreated(bool);

[VB] Protected Property ChildControlsCreated As Boolean

[JScript] protected function get ChildControlsCreated() : Boolean;protected function set ChildControlsCreated(Boolean);

Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a value that indicates whether the server control's child controls have been created.

ClientID

**ToString** 

ClientID {get;} string virtual [C#] public String\* get ClientID(); property virtual [C++]public: As String **Property** ClientID **Public** ReadOnly Overridable [VB] ClientID() String; public function get [JScript]

# Description

Gets the server control identifier generated by ASP.NET.

ASP.NET automatically generates a **ClientID** for a server control regardless of whether you have specified an **ID** property for it or not. A **ClilentID** is assigned to all **System.Web.UI.LiteralControl** objects (text and HTML that is not processed on the server) on a page when it is requested. You can view the **ClientID** s of each control in a page's control hierarchy by enabling tracing for the page or application. The format for automatically generated **ClientID** s is Ctrl0 for the first control on a page, Ctrl1 for the next control, Ctrl2 for the third control, and so on until all controls for the page are given a **ClientID** value.

Context

## **ToString**

[C#] protected

2

6

7

8

9

10

11

12

13

14

15

16

17

virtual

HttpContext

Context

{get;}

[C++]protected: property

virtual

HttpContext\* get Context();

[VB] Overridable Protected ReadOnly Property Context As HttpContext

get

function protected

Context()

HttpContext;

# Description

[JScript]

Gets the System. Web. HttpContext object associated with the server control for the current Web request.

This property gives you access to the Context object for the current Web request. The object provides properties that access the Application, Session, Request, Response, and other objects that contain information about the current HTTP request. It also provides methods that allow you to obtain configuration information and set or clear errors for the request.

Controls

**ToString** 

18

21

22

23

24

public [C#] 19 20

virtual

ControlCollection

Controls

{get;}

public: [C++]

virtual property [VB] Overridable Public ReadOnly Property Controls As ControlCollection

ControlCollection\*

get Controls();

[JScript]

public

function get Controls()

ControlCollection;

Description

25

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a **System.Web.UI.ControlCollection** object that represents the child controls for a specified server control in the UI hierarchy.

On an ASP.NET page, when controls are added declaratively between the opening and closing tags of a server control, ASP.NET automatically adds the controls to the containing server control's **System.Web.UI.ControlCollection**. Any HTML tags or text strings that are not processed on the server are compiled into **System.Web.UI.LiteralControl** objects. These are added to the collection like other server controls.

**EnableViewState** 

**ToString** 

**EnableViewState** {get; set;} virtual bool public [C#] [C++] public: \_\_property virtual bool get\_EnableViewState();public: \_\_property set EnableViewState(bool); virtual void Boolean EnableViewState As Overridable **Public Property** [VB] [JScript] public function get EnableViewState(): Boolean; public function set EnableViewState(Boolean);

# Description

Gets or sets a value indicating whether the server control maintains its view state, and the view state of any child controls it contains, when the current page request ends.

You must enable view state for the server control to maintain its state across HTTP requests.

**Events** 

# **ToString**

2

3

5

6

7

8

9 10

11

12 13

> 15 16

17 18

19

2021

22

2324

25

[C#] protected EventHandlerList Events {get;}

[C++] protected: \_\_property EventHandlerList\* get\_Events();

[VB] Protected ReadOnly Property Events As EventHandlerList

[JScript] protected function get Events() : EventHandlerList;

# Description

Gets a list of event handler delegates for the control. This property is readonly.

This property is of type **System.ComponentModel.EventHandlerList**, which uses a linear search algorithm to find entries in the list of delegates. A linear search algorithm is inefficient when working with a large number of entries. Therefore, when you have a large list, finding entries with this property will be slow.

HasChildViewState

**ToString** 

[C#] protected bool HasChildViewState {get;}

[C++] protected: \_\_property bool get\_HasChildViewState();

[VB] Protected ReadOnly Property HasChildViewState As Boolean

[JScript] protected function get HasChildViewState() : Boolean;

Description

Gets a value indicating whether the current server control's child controls 1 have any saved view-state settings. calls the avoid unnecessary to You can 3 System. Web. UI. Control. Clear Child View State method by using this property to 4 verify that any child controls of the server control are storing view-state 5 information. 6 ID 7 **ToString** 8 9 ID {get; set;} public virtual string [C#] 10 [C++] public: \_\_property virtual String\* get\_ID();public: \_\_property virtual void 11 set ID(String\*); 12 Property ID As String [VB] Overridable Public 13 [JScript] public function get ID() : String; public function set ID(String); 14 15 Description 16 Gets or sets the programmatic identifier assigned to the server control. 17 Setting this property on a server control provides you with programmatic 18 access to the server control's properties, events, and methods. This property can be 19 set by Web developers by declaring an id attribute in the opening tag of an 20 ASP.NET server control. 21 IsTrackingViewState 22 **ToString** 23 24 IsTrackingViewState {get;} protected bool [C#]

get IsTrackingViewState(); bool [C++]protected: property Boolean ReadOnly Property IsTrackingViewState As Protected [VB] Boolean; IsTrackingViewState() [JScript] protected function get

## Description

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a value that indicates whether ther server control is saving changes to its view state.

For a sample custom server control that uses this property, see .

NamingContainer

**ToString** 

NamingContainer {get;} Control virtual public [C#] Control\* get NamingContainer(); virtual [C++]public: property [VB] Overridable Public ReadOnly Property NamingContainer As Control NamingContainer() Control; public function [JScript] get

## Description

Gets a reference to the server control's naming container, which creates a unique namespace for differentiating between server controls with the same **System.Web.UI.Control.ID** property value.

Each page in an ASP.NET Web application contains a hierarchy of controls. This hierarchy is not dependent on whether a control generates UI visible to the user. The naming container for a given control is the parent control above it in the hierarchy that implements the **INamingContainer** interface. A server

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

control that implements this interface creates a unique namespace for the ID property values of its child server controls. Page **ToString** set;} Page {get; virtual Page public [C#] [C++] public: \_\_property virtual Page\* get\_Page();public: \_\_property virtual void set Page(Page\*); As Page Property Page Public Overridable [VB] [JScript] public function get Page() : Page;public function set Page(Page); Description Gets a reference to the System. Web. UI. Page instance that contains the server control. This property reflects the name of the .aspx file associated with the current Web request. In the control hierarchy, name of the page is \_PAGE by default. It is assigned to a type defined by the name of the .aspx file. Parent **ToString** Control Parent {get;} virtual public [C#] get Parent(); Control\* virtual public: property [C++]Control Public ReadOnly **Property** Parent As Overridable [VB] Control; Parent() public function get [JScript]

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets a reference to the server control's parent control in the page UI hierarchy.

Whenever a page is requested, a hierarchy of server controls on that page is built. This property allows you to determine the parent control of the current server control in that hierarchy, and to program against it.

Site

**ToString** 

Site set;} {get; **ISite** public [C#] get Site();public: property void ISite\* public: property [C++]set Site(ISite\*); **ISite** Site As

[VB] Public Property Site As ISite [JScript] public function get Site() : ISite; public function set Site(ISite);

# Description

Gets information about the Web site to which the server control belongs.

Sites bind a **Component** to a **Container** and enable communication between them. They also provide a way for the container to manage its components.

TemplateSourceDirectory

**ToString** 

[C#] public virtual string TemplateSourceDirectory {get;}

1	[C++] public:property virtual String* get_TemplateSourceDirectory();							
2	[VB] Overridable Public ReadOnly Property TemplateSourceDirectory As String							
3	[JScript] public function get TemplateSourceDirectory() : String;							
4								
5	Description							
6	Gets the virtual directory of the System.Web.UI.Page or							
7	System.Web.UI.UserControl that contains the current server control.							
8	UniqueID							
9	ToString							
10								
11	[C#] public virtual string UniqueID {get;}							
12	[C++] public:property virtual String* get_UniqueID();							
13	[VB] Overridable Public ReadOnly Property UniqueID As String							
14	[JScript] public function get UniqueID() : String;							
15								
16	Description							
17	Gets the unique, hierarchically-qualified identifier for the server control.							
18	This property differs from the ID property, in that the UniqueID property							
19	includes the identifier for the server control's naming container. This identifier is							
20	generated automatically when a page request is processed.							
21	ViewState							
22	ToString							
23								
24	[C#] protected virtual StateBag ViewState {get;}							
25	[C++] protected:property virtual StateBag* get_ViewState();							

[VB] Overridable Protected ReadOnly Property ViewState As StateBag [JScript] protected function get ViewState() : StateBag;

Description

Gets a dictionary of state information that allows you to save and restore the view state of a server control across multiple requests for the same page.

A server control's view state is the accumulation of all its property values. In order to preserve these values across HTTP requests, ASP.NET server controls use this property, which is an instance of the **StateBag** class, to store the property values. The values are then passed as a variable to a hidden field when subsequent requests are processed. For more information about view state, see . For more information about saving server control view state, see .

ViewStateIgnoresCase

**ToString** 

[C#] protected virtual bool ViewStateIgnoresCase {get;}
[C++] protected: \_\_property virtual bool get\_ViewStateIgnoresCase();
[VB] Overridable Protected ReadOnly Property ViewStateIgnoresCase As
Boolean

[JScript] protected function get ViewStateIgnoresCase() : Boolean;

Description

Gets a value that indicates whether the **System.Web.UI.StateBag** object is case-insensitive.

Visible

# **ToString**

virtual

bool

Visible

{get;

set;}

2

5

6

7 8

9

10

11 12

13

14

15 16

17

18 19

20 21

22

23

24

25

[C#] public [C++] public: \_\_property virtual bool get\_Visible();public: \_\_property virtual void set Visible(bool);

Boolean Visible As [VB] Public **Property** Overridable Boolean; public function set [JScript] public function get Visible() : Visible(Boolean);

#### Description

Gets or sets a value that indicates whether a server control is rendered as UI on the page.

Server controls that are not visible take up no space on a Web Forms page when it is displayed.

**ToString** 

DataBinding; EventHandler [C#] public event

DataBinding; EventHandler\* [C++]public: event

EventHandler DataBinding Public As Event [VB]

# Description

Occurs when the server control binds to a data source.

This event notifies the server control to perform any data binding logic that has been written for it.

**ToString** 

Disposed; EventHandler [C#] public event EventHandler\* Disposed; public: sealed event [C++]EventHandler Disposed Public As [VB] NotOverridable Event

## Description

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Occurs when a server control is released from memory, which is the last stage of the server control lifecycle when an ASP.NET page is requested.

Resources that require significant processor time, such as database connections, should be released with this event.

# **ToString**

public EventHandler Init; event [C#] EventHandler\* Init; [C++]public: event EventHandler Public Event Init As [VB]

## Description

Occurs when the server control is initialized, which is the first step in the its lifecycle.

Server controls should perform any initialization steps that are required to create and set up an instance. You cannot use view-state information within this event; it is not populated yet. You should not access another server control during this event, regardless of whether it is a child or parent to this control. Other server controls are not certain to be created and ready for access.

# **ToString**

ı D
ij
i' 5" 1
4,2
H
; =lfa
: 3t
ļ. da

[C#]	public	event		EventHandler	Load;
[C++]	public:	event		EventHandler*	Load;
[VB]	Public	Event	Load	As	EventHandler

Occurs when the server control is loaded into the System.Web.UI.Page object.

Notifies the server control to perform any processing steps that are set to occur on each page request. You can use view state information with this event. You can also access other server controls within the page's control hierarchy.

**ToString** 

[C#]	public	even	t EventH	Handler	PreRender;
[C++]	public:	eve	ent Eventl	Handler*	PreRender;
[VB]	Public	Event	PreRender	As	EventHandler

# Description

Occurs when the server control is about to render to its containing System. Web.UI.Page object.

Use this event to perform any updates before the output the server control is rendered to the page. Any changes in the view state of the server control can be saved during this event. Such changes made in the rendering phase will not be saved.

**ToString** 

[C#]	public	event	Eve	entHandler	Unload;
[C++]	public:	event	Ev	entHandler*	Unload;
[VB]	Public	Event	Unload	As	EventHandler

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Occurs when the server control is unloaded from memory.

Server controls must perform any final clean-up, such as closing files, database connections and discarding objects, during this stage of the control lifecycle before the instance is unloaded.

#### AddedControl

[C#] protected internal virtual void AddedControl(Control control, int index);
[C++] protected public: virtual void AddedControl(Control\* control, int index);
[VB] Overridable Protected Friend Dim Sub AddedControl(ByVal control As Control, ByVal index As Integer)
[JScript] package function AddedControl(control : Control, index : int);

#### Description

## AddParsedSubObject

AddParsedSubObject(object obi); void protected virtual [C#] protected: virtual void AddParsedSubObject(Object\* obj); [C++][VB] Overridable Protected Sub AddParsedSubObject(ByVal obj As Object) Object); AddParsedSubObject(obj function [JScript] protected

Notifies the server control that an element, either XML or HTML, was parsed, and adds the element to the server control's System.Web.UI.ControlCollection object.

Unless you override it, this method automatically adds System.Web.UI.LiteralControl objects to the server control's ControlCollection object. This collection is accessible through System.Web.UI.Control.Controls property. An System.Object that represents the parsed element.

#### BuildProfileTree

[C#] protected void BuildProfileTree(string parentId, bool calcViewState);
[C++] protected: void BuildProfileTree(String\* parentId, bool calcViewState);
[VB] Protected Sub BuildProfileTree(ByVal parentId As String, ByVal calcViewState As Boolean)

[JScript] protected function BuildProfileTree(parentId : String, calcViewState : Boolean);

#### Description

Gathers information about the server control and delivers it to the **System.Web.UI.Page.Trace** property to be displayed when tracing is enabled for the page.

This property gathers the information necessary about a page's UI hierarchy and passes it to the page's **Trace** property. When you enable tracing, either for a page or for your application, this information is displayed in the **Control Tree** 

section of the trace output. Trace output for a page is appended to the end of the page; while trace output for an application can be viewed from the trace viewer (trace.axd file) which is stored in the application's root directory. For more information about tracing, see . The identifier of the control's parent. A Boolean that indicates whether the view-state size is calculated.

#### ClearChildViewState

[C#]	protected	void	ClearChildViewState();
[C++]	protected:	void	ClearChildViewState();
[VB]	Protected	Sub	ClearChildViewState()
[JScript]	protected	function	ClearChildViewState();

## Description

Deletes the view-state information for all of the server control's child controls.

This method is commonly used when you override the System.Web.UI.Control.DataBind method when developing templated databound server controls. If you do not call this method, child control view-state information can be written to a parent server control, only to be overridden when data binding occurs.

#### CreateChildControls

[C#]	protected	virtual	void	CreateChildControls();
[C++]	protected:	virtual	void	CreateChildControls();
[VB]	Overridable	Protected	Sub	CreateChildControls()

1	THE THE
1	
1;	٥
÷	
	er E
ï	U
ľ	
;	375
i,	Ţ.
١,	ij
H	
;;	4
i;	:::
;,	ųį.
	ii.
ť	315 315
÷,	7
;1	
1	alla

[JScript] protected function CreateChildControls();

# Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Notifies server controls that use composition-based implementation to create any child controls they contain in preparation for posting back or rendering.

When you develop a composite or templated server control, you must override this method. For more information, see and .

## CreateControlCollection

ControlCollection CreateControlCollection(); virtual [C#] protected CreateControlCollection(); ControlCollection\* protected: virtual [C++]Function CreateControlCollection() As Overridable Protected [VB] ControlCollection

[JScript] protected function CreateControlCollection() : ControlCollection;

# Description

Creates a new **System.Web.UI.ControlCollection** object to hold the child controls (both literal and server) of the server control. *Return Value:* A **ControlCollection** object that contains the current server control's child server controls.

#### DataBind

[C#]	public	virtual	void	DataBind();
[C++]	public:	virtual	void	<pre>DataBind();</pre>
[VB]	Overridable	Public	Sub	DataBind()

lee**@**hayes piic 509-324-9256 1003 *MS1-863US.APP* 

		:	=	1111
			=	÷
	:		=	į
	:	***************************************	=	1
	.;	:	=	::
	ì,		=	å
		-		ii.
		11111		Series .
		::	;	Ę.
	١.		2	i
;	:	2 11 17 17		
	;			
:	;	:	:	:
;	ċ		:	į
:	;	4		
	•		1111	=
:	;	3.5	::	÷
÷	:	;	;	
;	;	ij	::	<u>.</u>
į	:	11 51 61 51	4	Ī
		3		

[JScript] public function DataBind();

Description

Binds a data source to the invoked server control and all of its child controls.

Use this method to bind data from a source to a server control. This method is commonly used after retrieving a data set through a database query.

## Dispose

[C#]	public	virtual	void	Dispose();
[C++]	public:	virtual	void	Dispose();
[VB]	Overridable	Public	Sub	Dispose()
[JScript]	public	fu	nction	Dispose();

# Description

Enables a server control to perform final clean up before it is released from memory.

Call **Dispose** when you are finished using the **System.Web.UI.Control**. The **Dispose** method leaves the **Control** in an unusable state. After calling this method, you must release all references to the control so the memory it was occupying can be reclaimed by garbage collection.

## EnsureChildControls

[C#]	protected	virtual	void	EnsureChildControls();
[C++]	protected:	virtual	void	EnsureChildControls();

ee⊗hayes plic 509+324-9256 1004 MS1-863US.APP

ū
ı, 🗇
13
ű
Ę
H
1 22
٠
1
[]
ļud

[VB] Overridable Protected Sub EnsureChildControls()

[JScript] protected function EnsureChildControls();

Description

2

3

4

5

6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Checks if the server control contains child controls. If it does not, it creates child controls.

If the server control contains nothing but literal content or HTML that is parsed and compiled as a **System.Web.UI.LiteralControl** object, it adds that object to the server control's **System.Web.UI.ControlCollection** object. The instance of the **LiteralControl** then becomes a child of the server control.

**FindControl** 

id); FindControl(string Control virtual public [C#] FindControl(String\* id); Control\* public: virtual [C++][VB] Overridable Public Function FindControl(ByVal id As String) As Control [JScript] public function FindControl(id : String) : Control; Searches the current control. specified for the server container naming

Description

Searches the current naming container for a server control with the specified *id* parameter.

\*Return Value: The specified control, or 0 if the specified control does not exist.

FindControl

The identifier for the control to be found.

[C#] protected virtual Control FindControl(string id, int pathOffset);
[C++] protected: virtual Control\* FindControl(String\* id, int pathOffset);
[VB] Overridable Protected Function FindControl(ByVal id As String, ByVal pathOffset As Integer) As Control
[JScript] protected function FindControl(id : String, pathOffset : int) : Control;

#### Description

1

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Searches the current naming container for a server control with the specified *id* and an integer, specified in the *pathOffset* parameter, that aids in the search.

Return Value: The specified control, or 0 if the specified control does not exist. The identifier for the control to be found. The number of controls up the page control hierarchy needed to reach a naming container.

#### HasControls

HasControls(); bool public virtual [C#] HasControls(); bool virtual public: [C++]As Boolean HasControls() **Function** Overridable **Public** [VB] HasControls() Boolean; function public [JScript]

## Description

Determines if the server control contains any child controls.

Return Value: true if the control contains other controls; otherwise, false.

lee@hayes pik 509-324-9256 1006 MS1-863US.APP

1 ||

2

3

4

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Since this method simply determines if any child controls exist, it can enhance performance by allowing you to avoid an unnecessary call to the **Count** property on the server control's **System.Web.UI.ControlCollection** object. The **ControlCollection** is available on the control through the **System.Web.UI.Control.Controls** property.

**IsLiteralContent** 

[C#] protected bool IsLiteralContent(); IsLiteralContent(); [C++]protected: bool **Function** IsLiteralContent() Boolean [VB] Protected As [JScript] function IsLiteralContent() Boolean; protected

#### Description

Determines if the server control holds only literal content.

Return Value: true if the server control is comprised of literal content; otherwise false.

When this method returns **true**, the server control's collection holds a single literal control. The content is then passed to the requesting browser as HTML.

#### LoadViewState

[C#] protected virtual void LoadViewState(object savedState); [C++]protected: virtual void LoadViewState(Object\* savedState); [VB] Overridable Protected Sub LoadViewState(ByVal savedState As Object) [JScript] protected function LoadViewState(savedState Object);

Restores view-state information from a previous page request that was saved by the **System.Web.UI.Control.SaveViewState** method.

Override this method when you need to customize how a custom server control restores its view state. For more information, see . An **System.Object** that represents the control state to be restored.

## MapPathSecure

[C#] protected string MapPathSecure(string virtualPath);
[C++] protected: String\* MapPathSecure(String\* virtualPath);
[VB] Protected Function MapPathSecure(ByVal virtualPath As String) As String
[JScript] protected function MapPathSecure(virtualPath : String) : String;

#### Description

Retrieves a mapped physical file path relative to the source file, if the requesting server control has sufficient security permissions to read the mapped result.

Return Value: The physical path to the requested file.

This method can only be used by server controls that have permissions to read files and which are part of fully trusted .dll files, such as System.Web.dll. This helps prevent security breaches. A relative or root relative URL.

#### OnBubbleEvent

[C#] protected virtual bool OnBubbleEvent(object source, EventArgs args);

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C++] protected: virtual bool OnBubbleEvent(Object\* source, EventArgs\* args);
[VB] Overridable Protected Function OnBubbleEvent(ByVal source As Object,
ByVal args As EventArgs) As Boolean
[JScript] protected function OnBubbleEvent(source : Object, args : EventArgs) :
Boolean;

Description

Determines whether the event for the server control is passed up the page's

UI server control hierarchy.

Return Value: true if the event has been cancelled; otherwise, false. The default is false.

server

controls

such

the

as

System.Web.UI.WebControls.DataList and System.Web.UI.WebControls.DataGrid Web controls can contain child controls that raise events. For example, each row in a DataGrid control can contain one or more buttons created dynamically by templates. The source of the event. An System.EventArgs object that contains the event data.

**OnDataBinding** 

ASP.NET

[C#] protected virtual void OnDataBinding(EventArgs e); OnDataBinding(EventArgs\* [C++]protected: virtual void e); [VB] Overridable Protected Sub OnDataBinding(ByVal e As EventArgs) function [JScript] protected OnDataBinding(e EventArgs);

lee **@**hayes pilc 509-324-9256 1009 *MS1-863US.APP* 

Raises the **DataBinding** event.

This method notifies a server control to perform any data binding logic that is associated with it. An **EventArgs** object that contains the event data.

OnInit

[C#] protected virtual void OnInit(EventArgs e);

[C++] protected: virtual void OnInit(EventArgs\* e);

[VB] Overridable Protected Sub OnInit(ByVal e As EventArgs)

[JScript] protected function OnInit(e : EventArgs);

## Description

Raises the **Init** event.

When notified by this method, server controls must perform any initialization steps that are required to create and set up an instance. In this stage of the server control's lifecycle, the control's view state has yet to be populated. Additionally, you can not access other server controls when this method is called either, regardless of whether it is a child or parent to this control. Other server controls are not certain to be created and ready for access. An **System.EventArgs** object that contains the event data.

OnLoad

[C#] protected virtual void OnLoad(EventArgs e);

[C++] protected: virtual void OnLoad(EventArgs\* e);

[VB] Overridable Protected Sub OnLoad(ByVal e As EventArgs)

[JScript] protected function OnLoad(e : EventArgs);

Description

2

3

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Raises the Load event.

This method notifies the server control that it should perform actions common to each HTTP request for the page it is associated with, such as setting up a database query. At this stage in the page lifecycle, server controls in the hierarchy are created and initialized, view state is restored, and form controls reflect client-side data. The **System.EventArgs** object that contains the event data.

OnPreRender

void OnPreRender(EventArgs [C#] protected virtual e); [C++]void OnPreRender(EventArgs\* protected: virtual e); [VB] Overridable Protected Sub OnPreRender(ByVal e As EventArgs) [JScript] protected function OnPreRender(e EventArgs);

Description

Raises the **System.Web.UI.Control.PreRender** event, which notifies the server control that

This method notifies the server control to perform any necessary prerendering steps prior to saving view state and rendering content. An **System.EventArgs** object that contains the event data.

OnUnload

lee❷hayes ptc 509-324-9256 1011 MS1-863US.APP

[C#] void OnUnload(EventArgs protected virtual e); protected: virtual void OnUnload(EventArgs\* [C++]e); [VB] Overridable Protected Sub OnUnload(ByVal e As EventArgs) OnUnload(e [JScript] protected function EventArgs); 6 Description Raises the Unload event. An System. EventArgs object that contains event 8 data. RaiseBubbleEvent 10 11 protected void RaiseBubbleEvent(object source, EventArgs [C#] [C++] protected: void RaiseBubbleEvent(Object\* source, EventArgs\* args); 13 [VB] Protected Sub RaiseBubbleEvent(ByVal source As Object, ByVal args As EventArgs) 15 [JScript] protected function RaiseBubbleEvent(source : Object, args : EventArgs); 17 Description 18 Assigns any sources of the event and its information to the control's parent. 19 ASP.NET controls such server as the 20 System.Web.UI.WebControls.Repeater 21 System.Web.UI.WebControls.DataList and 22 System.Web.UI.WebControls.DataGrid Web controls can contain child controls 23 that raise events. For example, each row in a DataGrid control can contain one or 24 25

3

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

more buttons created dynamically by templates. The source of the event. An System.EventArgs object that contains the event data.

RemovedControl

[C#] protected internal virtual void RemovedControl(Control control);
[C++] protected public: virtual void RemovedControl(Control\* control);
[VB] Overridable Protected Friend Dim Sub RemovedControl(ByVal control As Control)

[JScript] package function RemovedControl(control : Control);

Description

Render

[C#] protected virtual void Render(HtmlTextWriter writer); [C++]protected: virtual void Render(HtmlTextWriter\* writer); [VB] Overridable Protected Sub Render(ByVal writer As HtmlTextWriter) [JScript] function protected Render(writer HtmlTextWriter);

Description

Sends server control content to a provided System.Web.UI.HtmlTextWriter object, which writes the content to be rendered on the client.

When developing custom server controls, you can override this method to generate content for an ASP.NET page. For more information, see . The **HtmlTextWriter** object that receives the server control content.

- 13		1		$\sim$ 1	• 1	1	
· ·	an	А	or	' ' '	27	11	en
- 1/	$\sim$ 11	u	u	$\sim$ 1	111	ш	$\sim$ 11

[C#] protected virtual void RenderChildren(HtmlTextWriter writer): void RenderChildren(HtmlTextWriter\* [C++]protected: virtual writer); Overridable Protected [VB] Sub RenderChildren(ByVal writer As HtmlTextWriter)

[JScript] protected function RenderChildren(writer : HtmlTextWriter);

#### Description

2

8

9

10

11

12

13

15

16

17

18

19

20

21

22

23

24

Outputs the content of a server control's children to a provided **System.Web.UI.HtmlTextWriter** object, which writes the content to be rendered on the client.

This method uses the delegate specified in the System.Web.UI.Control.SetRenderMethodDelegate(System.Web.UI.Render Method) method to render any Active Server Pages (ASP) code on the page. If no ASP code exists on the page, this method renders any child controls for the server control. The HtmlTextWriter object that receives the child controls.

#### RenderControl

[C#] public RenderControl(HtmlTextWriter void writer); [C++]public: void RenderControl(HtmlTextWriter\* writer); [VB] Public RenderControl(ByVal Sub writer As HtmlTextWriter) [JScript] public function RenderControl(writer HtmlTextWriter);

## Description

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Outputs server control content to a provided System. Web.UI. HtmlTextWriter object, then checks if tracing is enabled for the containing page and retrieves trace information about the server control.

If a server control's **System.Web.UI.Control.Visible** property is set to **true**, this method checks to see if tracing is enabled for the page, and renders the server control content to the page. The **HtmlTextWriter** object that receives the control content.

#### ResolveUrl

public ResolveUrl(string [C#] string relativeUrl); public: String\* ResolveUrl(String\* relativeUrl); [C++][VB] Public Function ResolveUrl(ByVal relativeUrl As String) As String [JScript] public function ResolveUrl(relativeUrl String) String;

#### Description

Resolves a relative URL to the absolute URL where the page or user control associated with this request resides.

Return Value: The absolute URL.

This method uses the **System.Web.UI.Control.TemplateSourceDirectory** property to resolve to the absolute URL. The returned URL is for client use, and contains the session cookie if appropriate. The relative URL associated with the HTTP request.

#### SaveViewState

[C#] protected virtual object SaveViewState();

[C++]protected: virtual Object\* SaveViewState(); [VB] Overridable Protected Function SaveViewState() As Object [JScript] function SaveViewState() Object: protected

## Description

6

7

10

11

12

13

14

15

16

17

18

20

21

22

23

24

Saves any server control view-state changes that have occurred since the time the page was posted back to the server.

Return Value: Returns the server control's current view state. If there is no view state associated with the control, this method returns **null**.

View state is the accumulation of the values of a server control's properties. These values are automatically placed in the server control's **System.Web.UI.Control.ViewState** property, which is an instance of the **System.Web.UI.StateBag** class. This property's value is then persisted to a string object after the save state stage of the server control life cycle. For more information, see .

#### SetRenderMethodDelegate

[C#] public void SetRenderMethodDelegate(RenderMethod renderMethod);
[C++] public: void SetRenderMethodDelegate(RenderMethod\* renderMethod);
[VB] Public Sub SetRenderMethodDelegate(ByVal renderMethod As RenderMethod)

[JScript] public function SetRenderMethodDelegate(renderMethod : RenderMethod);

#### Description

lee@hayes pik 509-324-9256 1016 MS1-863US.APP

2

3

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Assigns an event handler delegate to render the server control and its content into its parent control.

This method is supplied for implementation purposes only; you should never call it directly. The information necessary to pass to the delegate so that it can render the server control.

IParserAccessor.AddParsedSubObject

[C#] void IParserAccessor.AddParsedSubObject(object obj);

[C++] void IParserAccessor::AddParsedSubObject(Object\* obj);

[VB] Sub AddParsedSubObject(ByVal obj As Object) Implements

IParserAccessor.AddParsedSubObject

[JScript] function IParserAccessor.AddParsedSubObject(obj: Object);

TrackViewState

virtual void TrackViewState(); [C#] protected TrackViewState(); [C++]protected: virtual void [VB] Overridable Protected Sub TrackViewState() [JScript] protected function TrackViewState();

#### Description

Causes tracking of view-state changes to the server control so they can be stored in the server control's **System.Web.UI.StateBag** object. This object is accessible through the **System.Web.UI.Control.ViewState** property.

Invoke this method when you develop templated data-bound controls. This method alerts ASP.NET to monitor changes to a server control's view state, which

is required when you override the **System.Web.UI.Control.DataBind** method. For more information, see .

ControlBuilder class (System.Web.UI)

TrackViewState

## Description

Supports the parser to build a control and the child controls it contains.

By default every control in a page is associated with a default ControlBuilder class. This class adds a child control to the Controls collection for every nested control that it encounters within custom control tags. Additionally, it adds literal controls for text between nested control tags. You can override this default behavior by defining your own custom control builder class. This is done by applying a control builder attribute to your control builder class as follows: [ControlBuilderAttribute(typeof(ControlBuilderType))]

ControlBuilder

Example Syntax:

TrackViewState

[C#] public ControlBuilder();
[C++] public: ControlBuilder();
[VB] Public Sub New()
[JScript] public function ControlBuilder();

ControlType

TrackViewState

ControlType public {get;} [C#] Type get ControlType(); public: Type\* property [C++]ReadOnly **Property** ControlType As Type [VB] Public function ControlType() public [JScript] Type; get 6 Description Gets the type for the control to be created. This can be a class type, an 8 interface type, a value type and so on. **FChildrenAsProperties** 10 **TrackViewState** 11 12 [C#] protected bool **FChildrenAsProperties** {get;} 13 get FChildrenAsProperties(); [C++]protected: \_property bool Protected ReadOnly Property FChildrenAsProperties As Boolean [VB] 15 [JScript] protected FChildrenAsProperties() Boolean; function get 16 17 Description 18 Checks if there are children controls to be parsed. 19 true if there are children controls to be parsed, false otherwise. 20 FIsNonParserAccessor 21 **TrackViewState** 22 23 [C#] protected bool FIsNonParserAccessor {get;} 24

property

bool

get FIsNonParserAccessor();

protected:

```
[VB] Protected ReadOnly Property FIsNonParserAccessor As
                                                                         Boolean
                                                                         Boolean;
    [JScript]
               protected
                           function
                                           FIsNonParserAccessor()
                                     get
3
    Description
          Checks if the control implements the System.Web.UI.IParseAccessor
5
    interface.
          This is a read only property. It is true if the controls implements the
7
    System. Web. UI. IParse Accessor interface, otherwise it returns false.
8
          HasAspCode
9
          TrackViewState
10
11
                   public
                                                    HasAspCode
    [C#]
                                     bool
                                                                            {get;}
12
    [C++]
                  public:
                                 property
                                                   bool
                                                               get HasAspCode();
13
             Public
                        ReadOnly
                                                  HasAspCode
                                                                          Boolean
    [VB]
                                     Property
                                                                   As
14
                                                                         Boolean;
    [JScript]
                 public
                            function
                                                 HasAspCode()
                                         get
15
16
    Description
17
          Gets a value indicating whether the control contains any Active Server
18
    Pages (ASP) code.
19
          ID
20
          TrackViewState
21
22
                  public
                                  string
    [C#]
                                                  ID
                                                               {get;
                                                                             set;}
23
    [C++]
                                    String*
             public:
                      property
                                              get_ID();public:
                                                                property
                                                                             void
24
    set_ID(String*);
```

```
[VB]
                   Public
                                                     ID
                                                                  As
                                                                              String
                                   Property
    [JScript] public function get ID(): String; public function set ID(String);
3
    Description
           Gets or sets the ID property for the control to be built.
5
           InDesigner
           TrackViewState
                                                        InDesigner
                    protected
                                        bool
                                                                               {get;}
    [C#]
                  protected:
                                                                   get InDesigner();
                                                       bool
    [C++]
                                      property
10
                           ReadOnly
                                                      InDesigner
                                                                     As
                                                                            Boolean
    [VB]
             Protected
                                         Property
11
                               function
                                                     InDesigner()
                                                                            Boolean;
    [JScript]
                 protected
                                            get
12
13
    Description
14
           Allows to change the behavior of control builders.
15
           NamingContainerType
16
           TrackViewState
17
18
    [C#]
                  public
                                 Type
                                               NamingContainerType
                                                                               {get;}
19
                public:
                                            Type*
                                                        get NamingContainerType();
    [C++]
                            property
20
    [VB]
             Public
                      ReadOnly
                                   Property
                                               NamingContainerType
                                                                                Type
    [JScript]
                public
                           function
                                              NamingContainerType()
                                       get
                                                                               Type;
22
23
    Description
           Gets the type assigned to the control's naming container.
25
```

Parser **TrackViewState** 2 3 **TemplateParser** [C#] protected Parser {get;} TemplateParser\* [C++]protected: get Parser(); property ReadOnly TemplateParser [VB] Protected **Property** Parser As [JScript] protected function Parser() TemplateParser; get 8 Description Gets the control builder parser. 10 TagName 11 **TrackViewState** 12 13 [C#] public string **TagName** {get;} 14 public: String\* [C++]property get TagName(); 15 Public ReadOnly **TagName** [VB] Property String As 16 [JScript] public function TagName() String; get 17 18 Description 19 Gets the tag name for the control to be built. 20 AllowWhitespaceLiterals 21 22 [C#] public virtual bool AllowWhitespaceLiterals(); 23 [C++]public: virtual bool AllowWhitespaceLiterals(); 24 Overridable Public Function AllowWhitespaceLiterals() As Boolean

[JScript] public function AllowWhitespaceLiterals() : Boolean;

Description

2

3

4

5

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Allows spaces within a control to be created as a LiteralControl object. If you create a custom control builder, you can override this method *Return Value:* true in all cases.

#### AppendLiteralString

public void AppendLiteralString(string [C#] virtual s); AppendLiteralString(String\* public: void [C++]virtual s); [VB] Overridable Public Sub AppendLiteralString(ByVal s As String) function AppendLiteralString(s [JScript] public String);

# Description

Adds literal content to a control. Literal content is any text that is passed through to the browser without being processed on the server. For example, any HTML elements and text between their opening and closing tags are literal content. The content to add to the control.

# AppendSubBuilder

[C#] public virtual void AppendSubBuilder(ControlBuilder subBuilder);
[C++] public: virtual void AppendSubBuilder(ControlBuilder\* subBuilder);
[VB] Overridable Public Sub AppendSubBuilder(ByVal subBuilder As ControlBuilder)

[JScript] public function AppendSubBuilder(subBuilder : ControlBuilder);

#### Description

3

5

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Adds builders to the **System.Web.UI.ControlBuilder** object for any child controls that belong to the control. The **ControlBuilder** object assigned to the child control.

#### CloseControl

[C#]	public	virtual	void	CloseControl();
[C++]	public:	virtual	void	CloseControl();
[VB]	Overridable	Public	Sub	CloseControl()
[JScript]	public	fun	ection	CloseControl();

### Description

Inserts the closing tag for the control.

## CreateBuilderFromType

[C#] public static ControlBuilder CreateBuilderFromType(TemplateParser parser, ControlBuilder parentBuilder, Type type, string tagName, string id, IDictionary attribs, int line, string sourceFileName); [C++] public: static ControlBuilder\* CreateBuilderFromType(TemplateParser\* parser, ControlBuilder\* parentBuilder, Type\* type, String\* tagName, String\* id, IDictionary\* attribs, int line, String\* sourceFileName); Shared Function CreateBuilderFromType(ByVal parser As TemplateParser, ByVal parentBuilder As ControlBuilder, ByVal type As Type, ByVal tagName As String, ByVal id As String, ByVal attribs As IDictionary,

ByVal line As Integer, ByVal sourceFileName As String) As ControlBuilder [JScript] public static function CreateBuilderFromType(parser: TemplateParser, parentBuilder: ControlBuilder, type: Type, tagName: String, id: String, attribs: IDictionary, line: int, sourceFileName: String): ControlBuilder;

## Description

Creates a **System.Web.UI.ControlBuilder** object for the specified tag. *Return Value*: The builder that is responsible for creating the control. The **System.Web.UI.TemplateParser** object responsible for parsing the control. The **System.Web.UI.ControlBuilder** object responsible for building the control. The type assigned to the control that the builder will create. The name of the tag to be built. This allows the builder to support multiple tag types. The ID attribute assigned to the control. The **System.Collections.IDictionary** object that holds all of the specified tags attributes. The source file line number for the specified control. The name of the source file from which the control is to be created.

### GetChildControlType

[C#] public virtual Type GetChildControlType(string tagName, IDictionary attribs);

[C++] public: virtual Type\* GetChildControlType(String\* tagName, IDictionary\* attribs);

[VB] Overridable Public Function GetChildControlType(ByVal tagName As String, ByVal attribs As IDictionary) As Type [JScript] public function GetChildControlType(tagName : String, attribs : IDictionary) : Type;

lee@hayes piik 509+324+9256 1025 MS1-863US.APP

### Description

1

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Obtains the Type for the current control's children. Return Value: If a default property exists, this method returns the default property builder. If the type is a collection, a collection builder; if the type is a template, a template builder. If none of the above, this method returns the type of the property, or **null** if there are no properties. The name of the current control's tag. An array of attributes contained in the current control.

### HasBody

public virtual bool HasBody(); [C#] HasBody(); [C++]public: virtual bool [VB] Overridable Public Function HasBody() As Boolean function HasBody() Boolean; [JScript] public

### Description

Determines if a control has both an opening and closing tag.

Return Value: true if the control has an opening and closing tag; otherwise, false.

#### HtmlDecodeLiterals

[C#] public virtual bool HtmlDecodeLiterals(); [C++]public: virtual bool HtmlDecodeLiterals(); Overridable HtmlDecodeLiterals() [VB] Public Function As Boolean public [JScript] function HtmlDecodeLiterals() Boolean;

ee ⊗hayes pic 509-324-9256 1026 MS1-863US.APP

#### Description

3

5

6

7

8

10

11

12

13

15

16

17

18

19

20

21

22

23

Checks if the literal string of an HTML control must be decoded.

Return Value: true if the HTML control literal string is to be decoded, false otherwise.

Init

[C#] public virtual void Init(TemplateParser parser, ControlBuilder parentBuilder, **IDictionary** Type type, string tagName, string id, attribs); [C++] public: virtual void Init(TemplateParser\* parser, ControlBuilder\* parentBuilder, Type\* type, String\* tagName, String\* id, IDictionary\* attribs); [VB] Overridable Public Sub Init(ByVal parser As TemplateParser, ByVal parentBuilder As ControlBuilder, ByVal type As Type, ByVal tagName As String, ByVal id As String, ByVal attribs As IDictionary) [JScript] public function Init(parser : TemplateParser, parentBuilder : ControlBuilder, type: Type, tagName: String, id: String, attribs: IDictionary);

#### Description

Initializes the control builder when a Web request is made. The System.Web.UI.TemplateParser object responsible for parsing the control. The System.Web.UI.ControlBuilder object responsible for building the control. The type assigned to the control that the builder will create. The name of the tag to be built. This allows the builder to support multiple tag types. The ID attribute assigned to the control. The System.Collections.IDictionary object that holds all of the specified tags attributes.

## NeedsTagInnerText

NeedsTagInnerText(); bool virtual [C#] public NeedsTagInnerText(); virtual bool public: [C++]Boolean NeedsTagInnerText() Overridable Public Function [VB] Boolean; NeedsTagInnerText() public function [JScript]

Description

2

3

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Determines if a control tag needs inner text. If so, the System.Web.UI.ControlBuilder.SetTagInnerText(System.String) method must be

Return Value: true if the control tag does not have any inner text; otherwise, false.

The default is false.

OnAppendToParentBuilder

[C#] public virtual void OnAppendToParentBuilder(ControlBuilder parentBuilder);

[C++] public: virtual void OnAppendToParentBuilder(ControlBuilder\* parentBuilder);

[VB] Overridable Public Sub OnAppendToParentBuilder(ByVal parentBuilder As ControlBuilder)

[JScript] public function OnAppendToParentBuilder(parentBuilder):

ControlBuilder);

Description

2

3

4

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Notifies the current control builder that it is being added to a parent control builder. The ControlBuilder object to which the current builder is added.

SetTagInnerText

[C#] public virtual void SetTagInnerText(String text);

void SetTagInnerText(String\* text); public: virtual [C++]String) SetTagInnerText(ByVal text As Overridable Public Sub [VB] String); SetTagInnerText(text function public [JScript]

## Description

Provides the builder with the inner text of the control tag. The text to be provided.

ControlBuilderAttribute class (System.Web.UI)
ToString

## Description

Specifies a custom **System.Web.UI.ControlBuilder** object for building a control within the ASP.NET parser. This class cannot be inherited.

This attribute prefixes the class to be used by the control builder at to create a custom control at run time.

**ToString** 

[C#] public static readonly ControlBuilderAttribute Default;
[C++] public: static ControlBuilderAttribute\* Default;

5

7

8

9

10

12

13

14

15

16

17

18

19

20

21

22

23

24

25

ControlBuilderAttribute [VB] ReadOnly Default As Shared Public ControlBuilderAttribute; Default [JScript] public static var Description Specifies the new System. Web. UI. Control Builder Attribute object. By default, the new object is set to null. This field is read-only. ControlBuilderAttribute Example Syntax: **ToString** builderType); ControlBuilderAttribute(Type public [C#] builderType); ControlBuilderAttribute(Type\* public: [C++]Sub New(ByVal builderType As Type) Public [VB] ControlBuilderAttribute(builderType Type); [JScript] public function Description Specifies the control builder for a custom control. The association of the control builder type with a custom control is obtained by preeceding the custom control builder class with the following attribute definition: [ControlBuilderAttribute(typeof(controlbuildertype))] control builder type BuilderType **ToString** BuilderType {get;} Type [C#] public

get BuilderType(); Type\* public: property [C++]BuilderType Type As Public ReadOnly **Property** [VB] BuilderType() Type; function public get [JScript] 3 4 Description 5 Gets the type assigned to the control. This property is read-only. 6 **TypeId** Equals 8 9 Equals(object obj); bool override public [C#] 10 Equals(Object\* obj); bool public: [C++][VB] Overrides Public Function Equals(ByVal obj As Object) As Boolean 12 [JScript] public override function Equals(obj : Boolean; Object) 13 14 Description 15 Determines the control builder type for the specified control. 16 GetHashCode 17 18 GetHashCode(); int override public [C#] 19 GetHashCode(); int public: [C++]20 **Function** GetHashCode() As Integer Public Overrides [VB] 21 int; public function GetHashCode() [JScript] override 22 23 Description 24 25

1 Returns the hash code for the control object. Return Value: int representing the control object hash code **IsDefaultAttribute** 3 public override IsDefaultAttribute(); [C#] bool [C++]public: bool IsDefaultAttribute(); [VB] Overrides Public **Function** IsDefaultAttribute() As Boolean [JScript] public override function IsDefaultAttribute() Boolean; 9 Description 10 **Determines** default builder. the is the control current 11 Return Value: true if the current is the default control builder. 12 ControlCollection class (System.Web.UI) 13 **ToString** 14 15 16 Description 17 Provides a collection container that enables ASP.NET server controls to 18 maintain a list of their child controls. 19 can any of the properties and methods of the access 20 ControlCollection class through the System.Web.UI.Control.Controls property. 21 Since the Control class is the base class for all ASP.NET server controls, all 22 server controls inherit this property. 23 ControlCollection 24 Example Syntax: 25

**ToString** 

2

[C#]

public:

public

ControlCollection(Control

owner);

[C++]

puone:

ControlCollection(Control\*

owner);

5 [VB]

7

9

10

11

12

13

14

ot] publ

Public

Sub New(ByVal

owner

As Control)

[JScript]

public function

ControlCollection(owner

Control);

8 Description

Initializes a new instance of the **System.Web.UI.ControlCollection** class for the parent server control specified in the *owner* parameter. The ASP.NET server control that the control collection is created for.

Count

**ToString** 

15

16

18

17

19

20

21

22

24

25

public int Count {get;} [C#] [C++]public: \_\_property get\_Count(); int [VB] Public ReadOnly Property Count As Integer [JScript] public function Count() get int;

Description

Gets the number of server controls in the **ControlCollection** object for the specified ASP.NET server control.

IsReadOnly

ToString

```
[C#]
                     public
                                                       IsReadOnly
                                       bool
                                                                               {get;}
2
    [C++]
                   public:
                                                      bool
                                                                   get IsReadOnly();
                                     property
3
                                                     IsReadOnly
    [VB]
              Public
                         ReadOnly
                                        Property
                                                                     As
                                                                             Boolean
    [JScript]
                  public
                             function
                                                    IsReadOnly()
                                                                            Boolean;
                                           get
5
6
    Description
7
           Gets a value indicating whether the ControlCollection object is read-only.
8
           IsSynchronized
9
           ToString
10
11
    [C#]
                    public
                                                     IsSynchronized
                                     bool
                                                                                {get;}
12
    [C++]
                  public:
                                   property
                                                   bool
                                                                get IsSynchronized();
13
                        ReadOnly
             Public
    [VB]
                                      Property
                                                   IsSynchronized
                                                                             Boolean
                                                                      As
14
                 public
    [JScript]
                            function
                                                 IsSynchronized()
                                                                            Boolean;
                                         get
15
16
    Description
17
           Gets a value indicating whether the ControlCollection is synchronized.
18
           Item
19
           ToString
20
21
    [C#]
              public
                           virtual
                                        Control
                                                     this[int
                                                                   index]
                                                                               {get;}
22
    [C++]
              public:
                           property
                                       virtual
                                                  Control*
                                                              get Item(int
                                                                              index);
23
    [VB] Overridable Public Default ReadOnly Property Item(ByVal index As
24
    Integer)
                                           As
                                                                             Control
```

ControlCollectionObject.Item(index); [JScript] returnValue = 2 Description 3 Gets a reference to the server control at the specified index location in the ControlCollection object. The location of the server control in the 5 ControlCollection . Owner 7 **ToString** 8 9 Control Owner [C#] protected {get;} 10 get Owner(); \_\_property Control\* [C++]protected: 11 [VB] Protected ReadOnly Property Owner Control As 12 [JScript] protected function get Owner() Control; 13 14 Description 15 which Gets ASP.NET control the the server to 16 System. Web. UI. Control Collection object belongs. 17 SyncRoot 18 **ToString** 19 20 [C#] public object SyncRoot {get;} 21 Object\* [C++]public: property get SyncRoot(); 22 ReadOnly Property Public [VB] SyncRoot Object As 23 [JScript] public function SyncRoot() Object; get 24 25

## Description

1

2

3

4

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Description

Gets an object that can be used to synchronize access to the collection of controls.

Add

Add(Control public virtual void child); [C#] public: virtual void Add(Control\* child); [C++]child Control) [VB] Overridable Public Sub Add(ByVal As function Add(child Control); [JScript] public

### Description

Adds the specified System. Web. UI. Control object to the collection.

The new control is added to the end of an ordinal index array. The control can be an instance of any ASP.NET server control, a custom server control you create, or a literal control. The **Control** object to add to the collection.

AddAt

AddAt(int [C#] public virtual void index, Control child); [C++]public: virtual void AddAt(int index, Control\* child); [VB] Overridable Public Sub AddAt(ByVal index As Integer, ByVal child As Control) [JScript] public function AddAt(index child Control); int,

lee@hayes ptc 509-324-9256 1036 MSJ-863U5,APP

Adds the specified **System.Web.UI.Control** object to the collection at the specified index location.

The added control can be an instance of any ASP.NET server control, a custom server control you create, or a literal control. The location in the array to add the child control. The **Control** object to add to the collection.

Clear

[C#]	public	virtual	void	Clear();
[C++]	public:	virtual	void	Clear();
[VB]	Overridable	Public	Sub	Clear()
[JScript]	public	f	unction	Clear();

### Description

Removes all controls from the current server control's System.Web.UI.ControlCollection object.

Use this method to empty a custom control's **ControlCollection** when you override the **System.Web.UI.Control.CreateChildControls** and **System.Web.UI.Control.DataBind** methods. Do this when you develop composite, templated controls or templated data bound controls. For more information, see .

#### Contains

[C#] public virtual bool Contains(Control c);
[C++] public: virtual bool Contains(Control\* c);
[VB] Overridable Public Function Contains(ByVal c As Control) As Boolean

4

5

7

8

9

10

11

12

13

15

16

17

18

19

20

21

22

23

24

Boolean; [JScript] public function Contains(c Control) Description Determines whether the specified server control is in the parent server System.Web.UI.ControlCollection object. control's Return Value: true if the specified server control exists in the collection; otherwise, false. The server control to search for in the collection. CopyTo public CopyTo(Array int index); [C#] void array, void CopyTo(Array\* int index); [C++]public: sealed array, [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As Integer) public CopyTo(array [JScript] function Array, index int); Description Copies the child controls stored in the System.Web.UI.ControlCollection object to an System.Array object, beginning at the specified index location in the Array. The Array to copy the child controls to. The zero-based relative index in array where copying begins. GetEnumerator [C#] **IEnumerator** public GetEnumerator(); [C++]public: sealed IEnumerator\* GetEnumerator();

As IEnumerator

NotOverridable Public Function GetEnumerator()

	=
١,	
1	Į
į	
ľ	ij
ľ,	
ľ,	
1	
;;	
,	dia
ľ	
ľ,	200 200 200 200 200 200 200 200 200 200

2

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

IEnumerator; GetEnumerator() public function [JScript] Description Retrieves an enumerator that can iterate through the ControlCollection object. Return Value: The enumerator to iterate through the collection. IndexOf IndexOf(Control value); virtual int public [C#] value); IndexOf(Control\* virtual int public: [C++][VB] Overridable Public Function IndexOf(ByVal value As Control) As Integer IndexOf(value int; Control) function public [JScript] Description Retrieves the index of a specified System. Web. UI. Control object in the collection. Return Value: The index of the specified server control. If the server control is not currently a member of the collection, it returns -1. Use this method to determine the index location of the specified server control in the ControlCollection object. The Control object for which the index is returned. Remove value); Remove(Control public virtual void [C#] value); Remove(Control\* void public: virtual

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

[VB] Overridable Public Sub Remove(ByVal value function Remove(value [JScript] public Description Removes the specified server control from the parent server control's System.Web.UI.ControlCollection object. index location, To remove a control from an System.Web.UI.ControlCollection.RemoveAt(System.Int32) server control to be removed. RemoveAt virtual void public [C#] void public: virtual [C++]Public Sub

RemoveAt(int index); RemoveAt(int index); index [VB] Overridable RemoveAt(ByVal As Integer) [JScript] public function RemoveAt(index int);

### Description

Removes a child control, at the specified index location, from the System.Web.UI.ControlCollection object.

To remove a control from the collection based on its value, use the  $System. Web. UI. Control Collection. Remove (System. Web. UI. Control) \\ \\ method.$ The ordinal index of the server control to be removed from the collection.

CssStyleCollection class (System.Web.UI)

**ToString** 

25

Control)

Control);

the

The

use

method.

As

## Description

Contains the HTML cascading-style sheets (CSS) inline style attributes for a specified HTML server control. This class cannot be inherited.

Any style declared for a particular HTML server control is added to the collection when the containing Web Forms page is parsed. It automatically parses and exposes CSS properties through a dictionary pattern API. You can manipulate any CSS property on a server control through the **System.Web.UI.HtmlControls.HtmlControl.Style** property. Simply use the CSS property's key and value in the indexed collection.

Count

**ToString** 

[C#]	pub	lic	int	Count		{get;}
[C++]	public	p:p	roperty	int	get	_Count();
[VB]	Public	ReadOnly	Property	Count	As	Integer
[JScript]	public	function	get	Count()	:	int;

# Description

Gets the number of items in the CssStyleCollection object.

Item

**ToString** 

[C#] public string this[string key] {get; set;}

3

4

5

6

7

8

9

10

11

12

13

16

17

18

19

20

21

22

23

24

```
[C++] public: __property String* get_Item(String* key);public: __property void
set Item(String*
                                          key,
                                                                       String*);
[VB] Public Default Property Item(ByVal key As String) As String
[JScript]
                                      returnValue
CssStyleCollectionObject.Item(key);CssStyleCollectionObject.Item(key)
returnValue;
Description
       Gets or sets a specified CSS value for the specified HTML server control.
The index to the CSS attribute.
       Keys
      ToString
[C#]
                public
                                  ICollection
                                                         Keys
                                                                          {get;}
[C++]
              public:
                             __property
                                                ICollection*
                                                                    get Keys();
[VB]
          Public
                     ReadOnly
                                     Property
                                                  Keys
                                                                    ICollection
                                                             As
[JScript]
              public
                          function
                                        get
                                                 Keys()
                                                                    ICollection;
Description
      Gets a collection of keys to all the styles in the CssStyleCollection object
for a specific HTML server control.
      Add
[C#]
          public
                       void
                                 Add(string
                                                  key,
                                                             string
                                                                        value);
[C++]
           public:
                       void
                                 Add(String*
                                                            String*
                                                  key,
                                                                        value);
```

[VB] Public Sub Add(ByVal key As String, ByVal value As String) public [JScript] function Add(key String, value String);

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Adds a style item to the CssStyleCollection object. The index assigned to the new style in the collection. The style to store in the collection.

Clear

[C#] public void Clear(); [C++]public: void Clear(); [VB] Public Sub Clear() [JScript] public function Clear();

Description

Removes all style items from the CssStyleCollection object.

Remove

[C#] public void Remove(string key); [C++]public: void Remove(String\* key); [VB] Public Sub Remove(ByVal key As String) [JScript] public function Remove(key String);

Description

Removes a style item from the CssStyleCollection object. The index of the style item to remove.

1 DataBinder class (System.Web.UI) **ToString** 2 3 4 Description 5 Provides design-time support for RAD designers to generate and parse . 6 This class cannot be inherited. 7 You 8 can use the overloaded System.Web.UI.DataBinder.Eval(System.Object,System.String) method of this class in Web Forms page data-binding syntax. This provides an easier syntax to 10 remember than standard data binding, but because DataBinder.Eval provides 11 automatic type conversion, can result in slower server response times. 12 DataBinder 13 Example Syntax: 14 **ToString** 15 16 [C#] public 17 DataBinder(); [C++]public: 18 DataBinder(); [VB] **Public** Sub 19 New() [JScript] public function DataBinder(); 20 Eval 21 22 [C#] public static object Eval(object 23 container, string expression); [C++] public: static Object\* Eval(Object\* container, String\* expression); 24 [VB] Public Shared Function Eval(ByVal container As Object, ByVal expression

As String) As Object [JScript] public static function Eval(container: Object, expression: String): Object; Uses reflection to parse and evaluate a data-binding expression against an object at runtime. This method allows RAD designers, such as Visual Studio.NET, to easily generate and parse data-binding syntax. This method can also be used declaratively on a Web Forms page to simplify casting to text that can be displayed in a browser.

Description

Evaluates data binding expressions at runtime.

Return Value: An System.Object that results from the evaluation of the data-binding expression.

While this method is automatically called when you create data bindings in a RAD designer, you can also use it declaratively if you want to simplify the casting to a text string to be displayed on a browser. To do so, you must place the tags, which are also used in standard ASP.NET data binding, around the data-binding expression. The object reference against which the expression is evaluated. This must be a valid object identifier in the page's specified language. The navigation path from the *container* to the property value to be placed in the bound control property. This must be a string type of property or field names separated by dots, such as **Table.DefaultView.Price**.

Eval

[C#] public static string Eval(object container, string expression, string format); [C++] public: static String\* Eval(Object\* container, String\* expression, String\* format);

[VB] Public Shared Function Eval(ByVal container As Object, ByVal expression
 As String, ByVal format As String) As String
 [JScript] public static function Eval(container : Object, expression : String, format
 : String)
 : String;

### Description

Evaluates data binding expressions at runtime and formats the output as text to be displayed in the requesting browser.

Return Value: A System.String that results from the evaluation of the data binding expression and conversion to a string type. This will be displayed by the requesting browser.

To learn more about format strings in the .NET Framework, see . The object reference against which the expression is evaluated. This must be a valid object identifier in the page's specified language. The navigation path from the *container* to the property value to be placed in the bound control property. This must be a string of property or field names separated by dots, such as **Table.DefaultView.Price** . A .NET Framework format string that converts the **System.Object** (which results from the evaluation of the data binding expression) to a **System.String** that can be displayed by the requesting browser.

# GetIndexedPropertyValue

[C#] public static object GetIndexedPropertyValue(object container, string expr); [C++] public: static Object\* GetIndexedPropertyValue(Object\* container, String\* expr);

[VB] Public Shared Function GetIndexedPropertyValue(ByVal container As Object, ByVal expr As String) As Object [JScript] public static function GetIndexedPropertyValue(container : Object, expr : String) : Object;

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

GetIndexedPropertyValue

[C#] public static string GetIndexedPropertyValue(object container, string propName, string format); [C++] public: static String\* GetIndexedPropertyValue(Object\* container, String\* String\* propName, format); [VB] Public Shared Function GetIndexedPropertyValue(ByVal container As Object, ByVal propName As String, ByVal format As String) As String [JScript] public static function GetIndexedPropertyValue(container : Object, propName String, format String) String:

Description

GetPropertyValue

[C#] public static object GetPropertyValue(object container, string propName);
 [C++] public: static Object\* GetPropertyValue(Object\* container, String\* propName);
 [VB] Public Shared Function GetPropertyValue(ByVal container As Object, ByVal propName As String) As Object

[JScript] public static function GetPropertyValue(container : Object, propName :

String) : Object;

| Description

1

2

3

4

5

6

7

8

9

10

11

12

13

15

16

17

18

19

20

21

22

23

24

25

GetPropertyValue

[C#] public static string GetPropertyValue(object container, string propName, string format); [C++] public: static String\* GetPropertyValue(Object\* container, String\* propName, String\* format); [VB] Public Shared Function GetPropertyValue(ByVal container As Object, ByVal propName As String, ByVal format As String) String As [JScript] public static function GetPropertyValue(container : Object, propName : String, format String) String;

Description

DataBinding class (System.Web.UI)

**ToString** 

Description

Contains information about a single data-binding expression in an ASP.NET server control, which allows rapid-application development (RAD) designers, such as Visual Studio .NET, to create data-binding expressions at design time. This class cannot be inherited.

14

19

21

24

25

3

4

5

7

8

Each data-binding expression in a server control is represented at design time by an instance of the **DataBinding** class. Any server control that contains one or more data-binding expressions has a System.Web.UI.DataBindingCollection object that contains the **DataBinding** objects. This collection is accessible by implementing the System. Web. UI. IData Bindings Accessor interface. Any DataBinding or DataBindingCollection objects associated with a server control exist only at design time. They do not exist at runtime and, therefore, are not accessible then.

**DataBinding** 

Example Syntax:

**ToString** 

[C#] public DataBinding(string propertyName, Type propertyType, string expression);

[C++] public: DataBinding(String\* propertyName, Type\* propertyType, String\* expression);

[VB] Public Sub New(ByVal propertyName As String, ByVal propertyType As ByVal expression As String) Type,

[JScript] public function DataBinding(propertyName : String, propertyType : expression String); Type,

Description

Initializes a new instance of the System. Web. UI. Data Binding class. The property to data bind to. The .NET Framework type of the property to data bind to. The data-binding expression to be evaluated.

1	Expression
2	ToString
3	
4	[C#] public string Expression {get; set;}
5	[C++] public:property String* get_Expression();public:property void
6	set_Expression(String*);
7	[VB] Public Property Expression As String
8	[JScript] public function get Expression() : String;public function set
9	Expression(String);
10	
11	Description
12	Gets or sets the data-binding expression to be evaluated.
13	Whether or not you bind data to a server control property, data-binding
14	expressions take the following form: .
15	PropertyName
16	ToString
17	
18	[C#] public string PropertyName {get;}
19	[C++] public:property String* get_PropertyName();
20	[VB] Public ReadOnly Property PropertyName As String
21	[JScript] public function get PropertyName() : String;
22	
23	Description
24	Gets the name of the ASP.NET server control property to be data bound
25	against.

3

5

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

PropertyType
ToString

[C#] public Type PropertyType {get;}

[C++] public: \_\_property Type\* get\_PropertyType();

[VB] Public ReadOnly Property PropertyType As Type

[JScript] public function get PropertyType() : Type;

Description

Gets the .NET Framework type of the data-bound ASP.NET server control property.

**Equals** 

Equals(object obj); override bool public [C#] Equals(Object\* obj); bool public: [C++][VB] Overrides Public Function Equals(ByVal obj As Object) As Boolean function Equals(obj Boolean; Object) [JScript] public override

Description

Determines whether the specified object is the same instance of the System.Web.UI.DataBinding class as the current object.

Return Value: true if the data binding property names match; otherwise, false.

The object to compare against the current DataBinding object.

GetHashCode

public override GetHashCode(); [C#] int public: int GetHashCode(); [C++][VB] Overrides Public **Function** GetHashCode() As Integer override function GetHashCode() int; [JScript] public 6 Description 7 Retrieves the hash code for an instance of the 8 object. System.Web.UI.DataBinding Return Value: A 32-bit signed integer hash code. 10 placed **DataBinding** objects in are a 11 System. Web. UI. Data Binding Collection object, accessible at design time through 12 the server control's System. Web. UI. Control. Data Bindings property, which is a 13 hash table that represents the bindings on a control. There can only be one binding 14 should property, the hashcode computation match the per SO 15 System.Web.UI.DataBinding.Equals(System.Object) implementation and only take the property name into account. 17 DataBindingCollection class (System.Web.UI) 18 **ToString** 19

21

20

22

23

Description

Provides a collection of **System.Web.UI.DataBinding** objects for an ASP.NET server control. This class cannot be inherited.

25

24

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

all of the table containing hash collection This is System. Web. UI. Data Binding objects on an ASP. NET server control. You can access the objects contained in this collection by implementing the System.Web.UI.IDataBindingsAccessor interface. **DataBinding** Any or DataBindingCollection objects associated with a server control exist only at design time. They do not exist at runtime and, therefore, are not accessible then.

DataBindingCollection

Example Syntax:

**ToString** 

```
[C#]publicDataBindingCollection();[C++]public:DataBindingCollection();[VB]PublicSubNew()[JScript]publicfunctionDataBindingCollection();
```

Description

Initializes a new instance of the System.Web.UI.DataBindingCollection class.

Count

**ToString** 

```
{get;}
                                                        Count
                                       int
                  public
[C#]
                                                                      get_Count();
                                     property
                                                        int
                 public:
[C++]
                                                                            Integer
                                                      Count
                                                                   As
           Public
                       ReadOnly
                                        Property
[VB]
                                                        Count()
                                                                                int;
                              function
                public
                                             get
[JScript]
```

lee@hayes pik 509-324-9256 1053 MS1-863US.APP

Description Gets the number of System. Web. UI. Data Binding objects in the 3 System. Web. UI. Data Binding Collection object. IsReadOnly 5 **ToString** 6 7 IsReadOnly {get;} bool [C#] public 8 get IsReadOnly(); bool public: property [C++]Boolean IsReadOnly As Property Public ReadOnly [VB] 10 Boolean; IsReadOnly() get [JScript] public function 11 12 Description 13 whether the indicating value Gets 14 System. Web. UI. Data Binding Collection is read-only. 15 IsSynchronized 16 **ToString** 17 18 IsSynchronized public {get;} bool [C#] 19 get IsSynchronized(); bool public: property [C++]20 Boolean IsSynchronized Property As Public ReadOnly [VB] 21 IsSynchronized() Boolean; public function get [JScript] 22 23 Description 24 25

indicating Gets value whether the System.Web.UI.DataBindingCollection is synchronized (thread-safe). 2 Item 3 **ToString** 5 [C#] public **DataBinding** this[string propertyName] {get;} 6 [C++] public: property DataBinding\* get Item(String\* propertyName); [VB] Public Default ReadOnly Property Item(ByVal propertyName As String) As **DataBinding** [JScript] returnValue DataBindingCollectionObject.Item(propertyName); 10 11 Description 12 Gets the System. Web. UI. Data Binding object with the specified property 13 name. The name of the property to be found. 14 RemovedBindings 15 **ToString** 16 17 public string[] RemovedBindings [C#] {get;} 18 [C++]public: property String\* get RemovedBindings(); 19 Public RemovedBindings [VB] ReadOnly Property As String 20 [JScript] public function RemovedBindings() get String[]; 21 22 Description 23 Gets an array of the names of the System. Web. UI. Data Binding objects 24 removed from the collection. 25

1	Sync	Root					
2	ToS1	tring					
3							
4	[C#]	public	o	bject	SyncR	oot	{get;}
5	[C++]	public:	prop	erty	Object*	get_Syr	ncRoot();
6	[VB]	Public Re	adOnly	Property	SyncRoo	ot As	Object
7	[JScript]	public	function	get	SyncRoot	() :	Object;
8							
9	Description	r					
10	Gets	s an object	that can	be used	to synchro	onize access	to the
11	System.W	eb.UI.DataBi	ndingColle	ction .			
12	Add	l					
13							1 ' 1' \
14	[C#]	public	void		.dd(DataBind		binding);
15	[C++]	public:	void	A	dd(DataBind		binding);
16	[VB]	Public Su	b Add(	ByVal	binding		aBinding)
17	[JScript]	public	function	Add(	(binding	: Data	Binding);
18							
19	Descriptio						
20	Ado	_			UI.DataBind		to the
21		eb.UI.DataBi	ndingColle	ection . The	data binding	g object to add	1.
22	Cle	ar					
23					• •		C1 ()-
24	[C#]		public		void		Clear();
25	[C++]		public:		void		Clear();

1,	
١,	
ı,	
	3
ľ	
ij	1
Į,	ar ad
1	
ij	
ť	an.
: . : .	
	ng Izlis
;	
• •	12112
i,	uni.
:	:dla

Clear() Sub [VB] Public Clear(); function public [JScript] 3 Description the System.Web.UI.DataBinding from objects all Removes 5 System.Web.UI.DataBindingCollection . 6 CopyTo 7 8 index); int void CopyTo(Array array, public [C#] 9 index); CopyTo(Array\* int array, void public: sealed [C++]10 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As 11 Integer) 12 int); index function CopyTo(array Array, public [JScript] 13 14 Description 15 **DataBindingCollection** values a one-dimensional to 16 System.Array, beginning at the Array object's specified index. The one-17 dimensional System.Array object that is the destination of the values copied from 18 DataBindingCollection. The index in the array, specified by the array parameter, 19 where copying begins. 20 GetEnumerator 21 22 GetEnumerator(); public **IEnumerator** [C#] 23 GetEnumerator(); sealed IEnumerator\* public: [C++]24 NotOverridable Public Function GetEnumerator() **IEnumerator** As [VB]

IEnumerator; GetEnumerator() [JScript] public function 2 Description 3 through the iterate to enumerator Returns an 4 object. System.Web.UI.DataBindingCollection 5 Return Value: An System.Collections.IEnumerator that contains the collection's members. 7 Remove 8 9 Remove(DataBinding binding); public void [C#] 10 binding); Remove(DataBinding\* public: void [C++]11 DataBinding) Remove(ByVal binding As Public Sub [VB] 12 DataBinding); function Remove(binding public [JScript] 13 14 Description 15 Removes the specified System. Web. UI. Data Binding object from the 16 System.Web.UI.DataBindingCollection . The System.Web.UI.DataBinding 17 object to be removed from the System. Web. UI. Data Binding Collection. 18 Remove 19 20 propertyName); Remove(string void public [C#] 21 Remove(String\* propertyName); public: void [C++]22 String) As propertyName Public Sub Remove(ByVal [VB] 23 public function Remove(propertyName : String); Removes a 24 from the object System.Web.UI.DataBinding

## System.Web.UI.DataBindingCollection

Description

Removes the **System.Web.UI.DataBinding** object associated with the specified property name from the **System.Web.UI.DataBindingCollection**. The property name associated with the **System.Web.UI.DataBinding** object to be removed.

Remove

[C#] public void Remove(string propertyName, bool addToRemovedList);
[C++] public: void Remove(String\* propertyName, bool addToRemovedList);
[VB] Public Sub Remove(ByVal propertyName As String, ByVal addToRemovedList As Boolean)
[JScript] public function Remove(propertyName : String, addToRemovedList : Boolean);

#### Description

Removes the System.Web.UI.DataBinding object, associated with the specified property name, from the System.Web.UI.DataBindingCollection and instructs the collection not to add the removed DataBinding object to the System.Web.UI.DataBindingCollection.RemovedBindings list. The property associated with the DataBinding object to be removed. A Boolean that indicates whether to add the property name to the RemovedBindings list.

DataBindingHandlerAttribute class (System.Web.UI)

**ToString** 

1 2 Description 3 Specifies a design-time class that performs databinding of controls within a designer. 5 **ToString** 6 7 Default; DataBindingHandlerAttribute readonly [C#] public static 8 DataBindingHandlerAttribute\* Default; static [C++]public: Shared ReadOnly Default As DataBindingHandlerAttribute Public [VB] 10 DataBindingHandlerAttribute; Default static [JScript] public var 11 12 Description 13 the for default constructor **Defines** the 14  ${\bf System. Web. UI. Data Binding Handler Attribute\ class.\ This\ field\ is\ read-only.}$ 15 DataBindingHandlerAttribute 16 Example Syntax: 17 **ToString** 18 19 DataBindingHandlerAttribute(); public [C#] 20 DataBindingHandlerAttribute(); public: [C++]21 Sub New() **Public** [VB] 22 public function DataBindingHandlerAttribute(); Initializes a new [JScript] 23 class. System.Web.UI.DataBindingHandlerAttribute of the instance 24 25

11	
1	
2	Description
3	Initializes a new instance of the
4	System.Web.UI.DataBindingHandlerAttribute class using no parameters. This
5	is the default constructor.
6	DataBindingHandlerAttribute
7	Example Syntax:
8	ToString
9	
10	[C#] public DataBindingHandlerAttribute(string typeName);
11	[C++] public: DataBindingHandlerAttribute(String* typeName);
12	[VB] Public Sub New(ByVal typeName As String)
13	[JScript] public function DataBindingHandlerAttribute(typeName : String);
14	
15	Description
16	Initializes a new instance of the
17	System.Web.UI.DataBindingHandlerAttribute class with the specified type
18	name.
19	The syntax for this attribute is: [DataBindingHandlerAttribute
20	typeof((dataBindingHandlerType))] The name of the data binding handler's type.
21	DataBindingHandlerAttribute
22	Example Syntax:
23	ToString
24	
25	[C#] public DataBindingHandlerAttribute(Type type);
1	

1	[C++]	publi	c: I	DataBinding	;HandlerAt	tribute(Type	*	type);
2	[VB]	Public	Sub	New(E	syVal	type	As	Type)
3	[JScript]	public	function	DataBind	ingHandler	Attribute(typ	pe :	Type);
4								
5	Description	on						
6	Ini	tializes	a	new	inst	ance	of	the
7	System.V	Veb.UI.Da	taBindingH	<b>IandlerAtt</b>	r <b>ibute</b> clas	s of the spec	cified Ty	ype. The
8	type name	e for this c	onstructor is	s the fully-q	ualified na	me of the typ	pe, inclu	iding the
9	assembly	name from	n which it v	vas loaded.	The Type	assigned to	the data	binding
10	handler.							
11	На	ındlerType	Name					
12	Те	String						
13								
14	[C#]	publi	c	string	Handle	erTypeName	;	{get;}
15	[C++]	public:	pro	perty	String*	get_Hand	ilerType	Name();
16	[VB]	Public	ReadOnly	Property	Handle	rTypeName	As	String
17	[JScript]	public	functio	n get	Handler	ГуреName()	:	String;
18								
19	Descripti							
20	G	ets the type	e name of t	he data bin	ding handl	er. If the typ	e name	is <b>null</b> ,
21	this prop	erty returns	s an empty s	tring.				
22	Ty	/peId						
23			iteralContro	ol class (Sys	tem.Web.U	Л)		
24	To	String						
25								

ClientID

2	
3	Description
4	Creates a control for HTML text that does not contain code processed on
5	the server, but persists the value of its
6	System.Web.UI.DataBoundLiteralControl.Text property to view state. This
7	class cannot be inherited.
8	This class allows the data binding syntax to be implemented.
9	DataBoundLiteralControl
10	Example Syntax:
11	ToString
12	
13	[C#] public DataBoundLiteralControl(int staticLiteralsCount, int
14	dataBoundLiteralCount);
15	[C++] public: DataBoundLiteralControl(int staticLiteralsCount, in
16	dataBoundLiteralCount);
17	[VB] Public Sub New(ByVal staticLiteralsCount As Integer, ByVal
18	dataBoundLiteralCount As Integer
19	[JScript] public function DataBoundLiteralControl(staticLiteralsCount : int
20	dataBoundLiteralCount : int)
21	
22	Description
23	
24	ChildControlsCreated

Context Controls								
Controls								
001111015								
EnableVi	iewState							
Events								
HasChild	lViewSta	te						
ID	ID							
IsTrackii	ngViewS	tate						
Naming(	Container							
Page								
Parent								
Site								
Template	eSourceI	Directory						
Text								
ToString	5							
Description								
Gets	or	sets	the	text	content	of	the	
System.Web.U	I.DataB	oundLite	ralContro	ol object.				
UniqueI	D							
ViewSta	ite							
ViewSta	iteIgnore	sCase						
Visible								
CreateC	ontrolCo	llection						
	Enable Vi Events Has Child ID Is Trackin Naming C Page Parent Site Template Text To String  Description Gets System. Web. U Unique I View State View State Visible	EnableViewState Events HasChildViewState ID IsTrackingViewState NamingContainer Page Parent Site TemplateSourceD Text ToString  Description Gets or System.Web.UI.DataB UniqueID ViewState ViewStateIgnore Visible	EnableViewState Events HasChildViewState ID IsTrackingViewState NamingContainer Page Parent Site TemplateSourceDirectory Text ToString  Description Gets or sets System.Web.UI.DataBoundLite UniqueID ViewState ViewStateIgnoresCase	EnableViewState Events HasChildViewState ID IsTrackingViewState NamingContainer Page Parent Site TemplateSourceDirectory Text ToString  Description Gets or sets the System.Web.UI.DataBoundLiteralControl UniqueID ViewState ViewStateIgnoresCase Visible	EnableViewState Events HasChildViewState ID IsTrackingViewState NamingContainer Page Parent Site TemplateSourceDirectory Text ToString   Description Gets or sets the text System.Web.UI.DataBoundLiteralControl object. UniqueID ViewState ViewStateIgnoresCase Visible	EnableViewState Events HasChildViewState ID IsTrackingViewState NamingContainer Page Parent Site TemplateSourceDirectory Text ToString  Description Gets or sets the text content System.Web.UI.DataBoundLiteralControl object. UniqueID ViewState ViewStateIgnoresCase Visible	EnableViewState Events HasChildViewState ID IsTrackingViewState NamingContainer Page Parent Site TemplateSourceDirectory Text ToString  Description Gets or sets the text content of System.Web.UI.DataBoundLiteralControl object. UniqueID ViewState ViewStateIgnoresCase Visible	

5

7

8

9

10

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#]	protected	override	ControlCollec	tion CreateCo	ntrolCollection	on();
[C++]	protect	ed: Co	ontrolCollection'	* CreateCo	ontrolCollection	on();
[VB]	Overrides	Protected	d Function	CreateControlC	Collection()	As
Contro	lCollection					
[JScrip	t] protect	ed overr	ide function	CreateContro	lCollection()	:
Contro	lCollection;					
Descrij	ption					
	Creates an	System.W	eb.UI.EmptyCo	llectionControl	object for	the
current	instance	of the	System.Web.	UI.DataBoundL	iteralContro	l.

LoadViewState

savedState); void LoadViewState(object protected override [C#] LoadViewState(Object\* savedState); void [C++]protected: [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object) [JScript] protected override function LoadViewState(savedState : Object);

Return Value: The EmptyCollectionControl object for the current control.

## Description

Loads the previously saved view state. You can override this method to synchronize System. Web. UI. Data Bound Literal Control. Text property with new literal content. The previously saved view state.

Render

1065 MS1-863US.APP lee@hayes plic 509+324+9256

[C#] protected override void Render(HtmlTextWriter output)
[C++] protected: void Render(HtmlTextWriter* output)
[VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter
[JScript] protected override function Render(output : HtmlTextWriter)
Description
Writes the content of the System.Web.UI.DataBoundLiteralContro
object to an output stream. The output stream that renders HTML content to the
client.
SaveViewState
[C#] protected override object SaveViewState(
[C++] protected: Object* SaveViewState(
[VB] Overrides Protected Function SaveViewState() As Object
[JScript] protected override function SaveViewState() : Object
Description
Saves any view state modified after the control began monitoring stat
changes.
Return Value: An System.Object that contains the savied view state values.
SetDataBoundString
[C#] public void SetDataBoundString(int index, string s
[C++] public: void SetDataBoundString(int index, String* s

H							~
1	[VB] Pub	lic Sub S	etDataBou	ndString(ByVal index A	s Integer, By	Val s As	String)
2	[JScript]	public	function	SetDataBoundString(inc	dex : int,	s :	String);
3							
4	Description	on					
5							
6	SetStaticString						
7							
8	[C#]	public	void	SetStaticString(int	index,	string	s);
9	[C++]	public:	void	SetStaticString(int	index,	String*	s);
10	[VB] Pub	olic Sub	SetStaticS	tring(ByVal index As I	nteger, ByV	al s As	String)
11	[JScript]	public	function	SetStaticString(index	: int,	s :	String)
12							
13	Description						
14							
15	De	signerDa	taBoundLi	teralControl class (Syste	m.Web.UI)		
16	Tra	ackViewS	State				
17							
18							
19	Description	on					
20	Si	mpler ver	rsion of Da	ntaBoundLiteralControlB	uilder, used	at design	time.
21	De	esignerDa	taBoundL	teralControl			
22	Ex	ample Sy	ntax:				
23	Tra	ackViewS	State				
24							
25	[C#]		public	Designe	erDataBound	LiteralC	ontrol()

DesignerDataBoundLiteralControl(); New() Sub DesignerDataBoundLiteralControl();

Description

1	Gets or sets the text content of the data-bound literal control.				
2	UniqueID				
3	ViewState				
4	ViewStateIgnoresCase				
5	Visible				
6	CreateControlCollection				
7					
8	[C#] protected override ControlCollection CreateControlCollection();				
9	[C++] protected: ControlCollection* CreateControlCollection();				
10	[VB] Overrides Protected Function CreateControlCollection() As				
11	ControlCollection				
12	[JScript] protected override function CreateControlCollection() :				
13	ControlCollection;				
14					
15	Description				
16					
17	LoadViewState				
18					
19	[C#] protected override void LoadViewState(object savedState);				
20	[C++] protected: void LoadViewState(Object* savedState);				
21	[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)				
22	[JScript] protected override function LoadViewState(savedState : Object);				
23					
24	Description				
25					

1	Loads the previously saved state. Overridden to synchronize Text property
2	with LiteralContent.
3	Render
4	
5	[C#] protected override void Render(HtmlTextWriter output);
6	[C++] protected: void Render(HtmlTextWriter* output);
7	[VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)
8	[JScript] protected override function Render(output : HtmlTextWriter);
9	
10	Description
11	Saves any state that was modified after the control began monitoring state
12	changes.
13	SaveViewState
14	
15	[C#] protected override object SaveViewState();
16	[C++] protected: Object* SaveViewState();
17	[VB] Overrides Protected Function SaveViewState() As Object
18	[JScript] protected override function SaveViewState() : Object;
19	
20	Description
21	The object that contains the state changes.
22	DesignTimeParseData class (System.Web.UI)
23	TrackViewState
24	DesignTimeParseData
25	Example Syntax:

**TrackViewState** 1 DataBindingHandler 2 **TrackViewState** 3 DesignerHost **TrackViewState** 5 DocumentUrl 6 TrackViewState 7 **ParseText** 8 **TrackViewState** 9 DesignTimeTemplateParser class (System.Web.UI) 10 **ToString** 11 ParseControl 12 13 ParseControl(DesignTimeParseData data); public static Control [C#] 14 ParseControl(DesignTimeParseData\* data); Control\* public: static [C++]15 [VB] Public Shared Function ParseControl(ByVal data As DesignTimeParseData) 16 Control As 17 [JScript] public static function ParseControl(data : DesignTimeParseData) : 18 Control; 19 ParseTemplate 20 21 ITemplate ParseTemplate(DesignTimeParseData static public 22 [C++] public: static ITemplate\* ParseTemplate(DesignTimeParseData\* As ParseTemplate(ByVal data Public Shared Function [VB] 24 ITemplate DesignTimeParseData) As

```
[JScript] public static function ParseTemplate(data : DesignTimeParseData) :
    ITemplate;
2
          EmptyControlCollection class (System.Web.UI)
3
           ToString
6
    Description
          Provides standard support for a ControlCollection that is always empty.
8
           This class is used when you want to define a custom control that does not
9
    allow child controls.
10
          EmptyControlCollection
11
           Example Syntax:
12
           ToString
13
14
    [C#]
                  public
                                   EmptyControlCollection(Control
                                                                             owner);
15
    [C++]
                                   EmptyControlCollection(Control*
                   public:
                                                                             owner);
    [VB]
               Public
                            Sub
                                     New(ByVal
                                                                             Control)
                                                                    As
                                                       owner
17
                                     EmptyControlCollection(owner
    [JScript]
                public
                          function
                                                                            Control);
18
19
    Description
20
           Instantiates an empty control collection.
21
           Count
22
           IsReadOnly
23
          IsSynchronized
24
           Item
25
```

1	Owner
2	SyncRoot
3	Add
4	
5	[C#] public override void Add(Control child);
6	[C++] public: void Add(Control* child);
7	[VB] Overrides Public Sub Add(ByVal child As Control)
8	[JScript] public override function Add(child : Control);
9	
10	Description
11	Denies the addition of the specified System.Web.UI.Control object to the
12	collection.
13	AddAt
14	
15	[C#] public override void AddAt(int index, Control child);
16	[C++] public: void AddAt(int index, Control* child);
17	[VB] Overrides Public Sub AddAt(ByVal index As Integer, ByVal child As
18	Control)
19	[JScript] public override function AddAt(index : int, child : Control);
20	
21	Description
22	Denies the addition of the specified System.Web.UI.Control object to the
23	collection, at the specified index position.
24	Html32TextWriter class (System.Web.UI)
25	ToString

2

3

4

6

7

8

9

10

11

12

13

14

16

17

18

19

20

22

23

24

25

Provides a text writer for ASP.NET server controls that render content to downlevel clients only.

Html32TextWriter

Example Syntax:

**ToString** 

System.Web.UI.Html32TextWriter

Description

Initializes a new instance of the **System.Web.UI.Html32TextWriter** class that uses the **System.Web.UI.HtmlTextWriter.DefaultTabString** constant when indentation of a line is necessary. The **System.IO.TextWriter** object to render the HMTL content.

Html32TextWriter

Example Syntax:

**ToString** 

[C#] public Html32TextWriter(TextWriter writer, string tabString);
[C++] public: Html32TextWriter(TextWriter\* writer, String\* tabString);
[VB] Public Sub New(ByVal writer As TextWriter, ByVal tabString As String)
[JScript] public function Html32TextWriter(writer : TextWriter, tabString : String);

lee@hayes plic 509+324+9256 1074 MS1-863US.APP

2

3

5

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the **System.Web.UI.Html32TextWriter** class using the specified tab spacing. The **System.IO.TextWriter** object to render the HMTL content. A **System.String** that represents the number of spaces defined in the **System.Web.UI.HtmlTextWriter.Indent** property.

Encoding

FontStack

**ToString** 

Description

Gets a collection of font information for the HTML to render.

FormatProvider

Indent

**InnerWriter** 

NewLine

**TagKey** 

TagName

GetTagName

[C#] protected override string GetTagName(HtmlTextWriterTag tagKey);

[C++] protected: String\* GetTagName(HtmlTextWriterTag tagKey);

[VB] Overrides Protected Function GetTagName(ByVal tagKey As

HtmlTextWriterTag) As String

lee@hayes pilc 509+324+9256 1075 *MS1-863US.APP* 

1	[JScript] protected override function GetTagName(tagKey : HtmlTextWriterTag) :
2	String;
3	
4	Description
5	Obtains the HTML element associated with the specified
6	System.Web.UI.HtmlTextWriterTag enumeration value.
7	Return Value: The HTML element.
8	If HtmlTextWriterTag.Div is passed in the tagKey parameter, this method
9	returns the HTML
10	element. The HtmlTextWriterTag value to obtain the HTML element for.
11	OnStyleAttributeRender
12	
13	[C#] protected override bool OnStyleAttributeRender(string name, string value,
14	HtmlTextWriterStyle key);
15	[C++] protected: bool OnStyleAttributeRender(String* name, String* value,
16	HtmlTextWriterStyle key);
17	[VB] Overrides Protected Function OnStyleAttributeRender(ByVal name As
18	String, ByVal value As String, ByVal key As HtmlTextWriterStyle) As Boolean
19	[JScript] protected override function OnStyleAttributeRender(name : String, value
20	: String, key : HtmlTextWriterStyle) : Boolean;
21	
22	Description
23	Determines whether the specified HTML style attribute and its value have
24	been rendered to the requesting page.
25	

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

Return Value: true if the HTML style attribute and its value have been rendered to the requesting page; otherwise, false.

This method supports a smaller number of HTML style attributes than the of the System.Web.UI.HtmlTextWriter.OnStyleAttributeRender(System.String,System.String,System.Web.UI.HtmlTextWriterStyle) method which it overrides. The HTML style attribute to render to the client. The value associated with the HTML style attribute. The System.Web.UI.HtmlTextWriterStyle enumeration value associated with the HTML style attribute.

OnTagRender

[C#] protected override bool OnTagRender(string name, HtmlTextWriterTag key);
[C++] protected: bool OnTagRender(String\* name, HtmlTextWriterTag key);
[VB] Overrides Protected Function OnTagRender(ByVal name As String, ByVal key As HtmlTextWriterTag) As Boolean
[JScript] protected override function OnTagRender(name : String, key : HtmlTextWriterTag) : Boolean;

# Description

Determines whether the specified HTML element has been rendered to the requesting page.

Return Value: true if the HTML element has been rendered to the requesting page; otherwise false.

This method associates the

2

6

10

11

13

14

15

16

17

18

19

20

21

22

23

24

25

element with an HTML table to simplify page layout for browsers that do not support this element. The HTML element to render. The **System.Web.UI.HtmlTextWriterTag** enumeration value associated with the HTML element.

## RenderAfterContent

[C#] protected override string RenderAfterContent(); String\* RenderAfterContent(); [C++]protected: Function [VB] Overrides Protected RenderAfterContent() String [JScript] protected override function RenderAfterContent() String;

## Description

Writes any text or spacing that occurs after the content of the HTML element to the **System.Web.UI.HtmlTextWriter** output stream. Return Value: The spacing or text to write prior to the content of the HTML element. If there is no such information to render, this method returns **null**.

#### RenderAfterTag

[C#] protected override RenderAfterTag(); string String\* RenderAfterTag(); [C++]protected: [VB] Overrides Protected **Function** RenderAfterTag() As String [JScript] protected override function RenderAfterTag() String;

## Description

lee@hayes plic 509432449256 1078 MS1-863US.APP

5

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Writes any spacing or text that occurs after an HTML element's closing tag.

Return Value: The spacing or text to write after the closing tag of the HTML element. If there is no such information to render, this method returns null.

#### RenderBeforeContent

RenderBeforeContent(); [C#] protected override string String\* [C++]protected: RenderBeforeContent(); Overrides Protected Function RenderBeforeContent() [VB] String function RenderBeforeContent() [JScript] protected override String:

# Description

Writes any tab spacing or font information that appears before the content contained in an HTML element.

Return Value: The font information or spacing to write prior to the content of the HTML element. If there is no such information to render, this method returns null

# RenderBeforeTag

RenderBeforeTag(); [C#] protected override string [C++]String\* RenderBeforeTag(); protected: [VB] Overrides Protected **Function** RenderBeforeTag() As String [JScript] override function protected RenderBeforeTag() String;

# Description

lee@hayes pilc 509-324-9256 1079

3

5

б

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Writes any text or tab spacing that occurs before the opening tag of an HTML element to the **System.Web.UI.HtmlTextWriter** output stream. *Return Value:* Any HTML font and spacing information to render before the tag; if there is no such information to render, this method returns **null**.

# RenderBeginTag

[C#] public override void RenderBeginTag(HtmlTextWriterTag tagKey); public: void [C++]RenderBeginTag(HtmlTextWriterTag tagKey); [VB] Overrides Public Sub RenderBeginTag(ByVal tagKey As HtmlTextWriterTag)

[JScript] public override function RenderBeginTag(tagKey: HtmlTextWriterTag);

## Description

Writes the opening tag of the specified HTML element to the System.Web.UI.HtmlTextWriter output stream.

#### If the

element is specified, this method performs basic HTML

formatting to present the HTML content. The System.Web.UI.HtmlTextWriterTag enumeration value that indicates the HTML element to write.

## RenderEndTag

[C#]	public	override	void	<pre>RenderEndTag();</pre>
[C++]	public:		void	RenderEndTag();
[VB]	Overrides	Public	Sub	RenderEndTag()

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[JScript] public override function RenderEndTag(); Description Writes the end of HTML element the tag an to System.Web.UI.HtmlTextWriter output stream along with any font information

HtmlTextWriter class (System.Web.UI)

WriteStyleAttribute

that is associated with the element.

# Description

Represents a writer that writes a sequential series of HTML-specific characters and text. This class provides formatting capabilities that ASP.NET server controls use when rendering HTML content to clients.

This class is commonly used to render HTML to a Web Forms page. Use it as an output stream when you override the System.Web.UI.Control.Render(System.Web.UI.HtmlTextWriter) , System.Web.UI.LiteralControl.Render(System.Web.UI.HtmlTextWriter) , and other ASP.NET server control methods that write content to a Web page.

WriteStyleAttribute

# Description

Represents the characters that make up the default tab.

WriteStyleAttribute

1											
2	[C#]	public	const	char	Double(	QuoteChar;					
3	[C++]	public:	const	wchar_t	Double(	QuoteChar;					
4	[VB]	Public	Const	DoubleQuoteChar	As	Char					
5	[JScript]	public	var	DoubleQuoteChar	:	Char;					
6											
7	Description										
8	Rep	resents a doub	le-quote chara	cter.							
9	Wri	teStyleAttribut	e								
10											
11	[C#]	public	const	string	EndTag	LeftChars;					
12	[C++]	public:	const	String*	EndTag	LeftChars;					
13	[VB]	Public	Const	EndTagLeftChars	As	String					
14	[JScript]	public	var	EndTagLeftChars	:	String;					
15											
16	Description	ı									
17	Rep	resents the left	angle bracket	t and closing forward s	lash for a	closing tag					
18	of an HTM	L element.									
19	Wri	teStyleAttribut	e								
20											
21	[C#]	public	con	est char	Е	qualsChar;					
22	[C++]	public:	const	wchar_t	Е	qualsChar;					
23	[VB]	Public	Const	EqualsChar	As	Char					
24	[JScript]	public	var	EqualsChar	:	Char;					
25											

Represents the equal sign character.

WriteStyleAttribute

[C#]	public	const	string F	EqualsDouble	eQuoteString;
[C++]	public:	const	String* E	EqualsDouble	eQuoteString;
[VB]	Public	Const	EqualsDoubleQuoteS	tring A	As String
[JScript]	public	var	EgualsDoubleQuote	String	: String;

# Description

Represents an equal sign, a forward slash, and a double quote character together in a **System.String** .

WriteStyleAttribute

[C#]	public	const	string	SelfClo	singChars;
[C++]	public:	const	String*	SelfClo	singChars;
[VB]	Public	Const	SelfClosingChars	As	String
[JScript]	public	var	SelfClosingChars	:	String;

# Description

Represents the self-closing forward slash character of an HTML tag.

This charcter is used in HTML elements that are self-closed. For example, .

WriteStyleAttribute

	[C#]	public	const	string	SelfClosingTagEn	ıd;					
	[C++]	public:	const	String*	SelfClosingTagEn	ıd;					
	[VB]	Public	Const	SelfClosingTagEnd	As Stri	ng					
	[JScript]	public	var	SelfClosingTagEnd	: Strin	ıg;					
	Description  Represents the closing forward slash and right angle bracket of a self-closing HTML element.  WriteStyleAttribute										
	[C#]	public	cons	t char	SemicolonCh	ar;					
	[C++]	public:	const	wchar_t	SemicolonCh	ar;					
	[VB]	Public	Const	SemicolonChar	As Cl	nar					
	[JScript]	public	var	SemicolonChar	: Ch	ar;					
	Description  Represents the semicolon character.  WriteStyleAttribute										
	[C#]	public	const	char	SingleQuoteCh	ar;					
	[C++]	public:	const	wchar_t	SingleQuoteCh	ar;					
	[VB]	Public	Const	SingleQuoteChar	As Ch	nar					
	[JScript]	public	var	SingleQuoteChar	: Ch	ar;					
1											

				•								
	1											
	2	Description										
	3	Repre	sents a single of	quote character.								
	4	Write	WriteStyleAttribute									
	5											
	6	[C#]	public	const	char		SlashChar;					
	7	[C++]	public:	const	_wchar_t		SlashChar;					
	8	[VB]	Public	Const	SlashChar	As	Char					
	9	[JScript]	public	var	SlashChar	:	Char;					
., 100	10											
	11	Description										
	12	Represents the backslash character.										
	13	WriteStyleAttribute										
	14											
	15	[C#]	public	const	char	;	SpaceChar;					
	16	[C++]	public:	const	wchar_t	i	SpaceChar;					
	17	[VB]	Public	Const	SpaceChar	As	Char					
	18	[JScript]	public	var	SpaceChar	:	Char;					
	19											
	20	Description										
	21	Repre	esents a space of	character.								
	22	Write	StyleAttribute									
	23											
	24	[C#]	public	const	char	StyleE	EqualsChar;					
	25	[C++]	public:	const	wchar_t	StyleE	EqualsChar;					

1	[VB]	Public	Const	StyleEqualsChar	As	Char					
2	[JScript]	public	var	StyleEqualsChar	:	Char;					
3											
4	Description	ı									
5	Represents the colon character.										
6	WriteStyleAttribute										
7											
8	[C#]	public	const	char	Tagl	LeftChar;					
9	[C++]	public:	const	wchar_t	Tagl	LeftChar;					
10	[VB]	Public	Const	TagLeftChar	As	Char					
11	[JScript]	public	var	TagLeftChar	:	Char;					
12											
13	Description	n									
14	Rep	resents the ope	ning angle-brac	eket character of an H	TML tag.						
15	Wri	teStyleAttribute	e								
16											
17	[C#]	public	const	char	•	ightChar;					
18	[C++]	public:	const	wchar_t	_	ightChar;					
19	[VB]	Public	Const	TagRightChar	As	Char					
20	[JScript]	public	var	TagRightChar	•	Char;					
21	_										
22	Description										
23	_		sing angle-bracl	ket character of an H7	IML tag.						
24		nlTextWriter									
25	Exa	mple Syntax:									

## WriteStyleAttribute

public

public:

2

1

3 [C#]

[C++]

5

7

8

9

10

11

12

14 15

16

17

19

18

20 21

22

23

2425

HtmlTextWriter(TextWriter writer);

HtmlTextWriter(TextWriter\* writer);

[VB] Public Sub New(ByVal writer As TextWriter)

[JScript] public function HtmlTextWriter(writer : TextWriter); Initializes a new

instance of the System.Web.UI.HtmlTextWriter class.

# Description

Initializes a new instance of the **System.Web.UI.HtmlTextWriter** class that uses the **System.Web.UI.HtmlTextWriter.DefaultTabString** constant when indentation of a line is necessary. The **System.IO.TextWriter** object to render the HMTL content.

HtmlTextWriter

Example Syntax:

WriteStyleAttribute

[C#] public HtmlTextWriter(TextWriter writer, string tabString);
[C++] public: HtmlTextWriter(TextWriter\* writer, String\* tabString);
[VB] Public Sub New(ByVal writer As TextWriter, ByVal tabString As String)
[JScript] public function HtmlTextWriter(writer : TextWriter, tabString : String);

## Description

Initializes a new instance of the **System.Web.UI.HtmlTextWriter** class with the line indenation as specified in the *tabString* parameter. The

1	System	ı.IO.TextWı	r <b>iter</b> obje	ect to rend	er the I	HMTL co	ontent. A Sy	ystem	.String that
2	represe	ents the	nun	nber	of	spaces	defined	i	in the
3	System	ı.Web.UI.Hı	tmlTextV	Vriter.Inc	lent pro	operty.			
4		Encoding							
5		WriteStyleA	ttribute						
6									
7	[C#]	public	ov	erride	Enc	coding	Encod	ling	{get;}
8	[C++]	public:	pro	perty	virtual	Enc	oding*	get_	Encoding();
9	[VB]	Overrides	Public	ReadOn	ly Pr	operty	Encoding	As	Encoding
10	[JScrip	t] publi	c fu	nction	get	Enco	ding()	:	Encoding;
11									
12	Descri	otion							
13		Gets	the	Syste	em.Tex	t.Encodi	ng	that	the
14	Systen	ı.Web.UI.H	tmlTextV	<b>Vriter</b> obj	ect use	s to write	content to	the pa	age.
15		FormatProvi	der						
16		Indent							
17		WriteStyleA	ttribute						
18									
19									
20	Descri	ption							
21		Gets or sets t	the numb	er of space	es to in	dent at th	e beginning	g of a	line.
22		InnerWriter							
23		WriteStyleA	ttribute						
24									
25	[C#]	public	Te	extWriter		InnerWri	ter	{get;	set;}

1	[C++] public:property TextWriter* get_InnerWriter();public:property void									
2	set_InnerWriter(TextWriter*);									
3	[VB] Public Property InnerWriter As TextWriter									
4	[JScript] public function get InnerWriter() : TextWriter; public function set									
5	InnerWriter(TextWriter);									
6										
7	Description									
8	Gets the text writer that writes the inner content of the HTML element.									
9	NewLine									
10	WriteStyleAttribute									
11										
12	[C#] public override string NewLine {get; set;}									
13	[C++] public:property virtual String* get_NewLine();public:property virtual									
14	void set_NewLine(String*);									
15	[VB] Overrides Public Property NewLine As String									
16	[JScript] public function get NewLine() : String;public function set									
17	NewLine(String);									
18										
19	Description									
20	Gets or sets the line terminator string used by the current									
21	System.Web.UI.HtmlTextWriter.									
22	The default line terminator string is a carriage return followed by a line feed									
23	("\r\n").									
24	TagKey									
25	WriteStyleAttribute									

AddAttribute

[C#]	protected	HtmlTextWr	iterTag	TagKey	{get;	set;}				
[C++]	protected:r	property Htn	nlTextWriter	Tag get	_TagKey();p	rotected:				
prop	property void set_TagKey(HtmlTextWriterTag);									
[VB]	Protected	Property	TagKey	As	HtmlTextV	VriterTag				
[JScrip	t] protected functi	on get TagKe	y() : HtmlTe	extWriterT	ag;protected	function				
set				TagKey(	HtmlTextWi	riterTag);				
Descrip	otion									
•	Gets or sets the System.Web.UI.HtmlTextWriterTag value for the									
specifie	specified HTML element.									
,	TagName									
7	WriteStyleAttribu	te								
[C#]	protected	string	TagN	lame	{get;	set;}				
[C++]	protected:prop	erty String*	get_TagNan	ne();protec	ted:prop	erty voic				
set_Tag	gName(String*);									
[VB]	Protected	Property	Tag	Name	As	String				
[JScrip	t] protected fun	ction get Ta	gName() :	String;pro	otected fund	ction set				
TagNaı	me(String);									
Descrip	otion									
(	Gets or sets the n	ame of the H	ΓML elemei	nt to be w	ritten to the	rendered				
nage										

21

23

25

[C#] public virtual void AddAttribute(HtmlTextWriterAttribute key, string value); [C++] public: virtual void AddAttribute(HtmlTextWriterAttribute key, String\* value);

[VB] Overridable Public Sub AddAttribute(ByVal key As HtmlTextWriterAttribute, ByVal value As String)

[JScript] public function AddAttribute(key : HtmlTextWriterAttribute, value : String);

## Description

Adds the HTML attribute associated with the specified System.Web.UI.HtmlTextWriterAttribute key and the specified value to the System.Web.UI.HtmlTextWriter output stream. An HtmlTextWriterAttribute value that represents the HTML attribute. The value to assign to the HTML attribute.

## AddAttribute

[C#] public virtual void AddAttribute(string string value); name, [C++] public: virtual void AddAttribute(String\* String\* name, value); [VB] Overridable Public Sub AddAttribute(ByVal name As String, ByVal value As String) [JScript] public function AddAttribute(name : String, value : String); Adds an HTML attribute and its value to an System.Web.UI.HtmlTextWriter output stream be rendered client. to on a

1

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Adds the specified HTML attribute and value to the HtmlTextWriter output stream. The HTML attribute to add. The value to assign to the HTML attribute.

#### AddAttribute

[C#] public virtual void AddAttribute(HtmlTextWriterAttribute key, string value, fEncode); bool [C++] public: virtual void AddAttribute(HtmlTextWriterAttribute key, String\* bool fEncode); value, [VB] Overridable Public Sub AddAttribute(ByVal As HtmlTextWriterAttribute, ByVal value As String, ByVal fEncode As Boolean) [JScript] public function AddAttribute(key: HtmlTextWriterAttribute, value: String, **f**Encode Boolean);

#### Description

Adds the HTML attribute associated with the System.Web.UI.HtmlTextWriterAttribute value specified by the key parameter, the specified attribute value, and a value that indicates if the attribute and value should be HTML encoded to the System.Web.UI.HtmlTextWriter output stream. An HtmlTextWriterAttribute value that represents the HTML attribute. The value to assign to the HTML attribute. true to HTML encode the attribute and its value; otherwise, false.

#### AddAttribute

[C#] pu	blic vir	tual voi	d Add	Attribute	(string name,	string	value, bo	ool fEnd	ode);
[C++] ]	public:	virtual	void	AddAttr	ibute(String*	name,	String*	value,	bool
fEndode	;								
[VB] Overridable Public Sub AddAttribute(ByVal name As String, ByVal value									
As	Stri	ing,	В	yVa1	fEndode		As	Boo	olean)
[JScript]	public	function	on Ado	dAttribute	(name : Strin	ng, valu	e : Strin	g, fEnd	ode:
Boolean	);								

# Description

Adds the specified HTML attribute and value to the HtmlTextWriter output stream, with a value that indicates if the attribute and value should be HTML encoded. The HTML attribute to add. The value to assign to the HTML attribute. true to HTML encode the attribute and its value; otherwise, false.

## AddAttribute

[C#]	protected	virtual	void	AddAttribu	ute(string	name,	string	value,			
HtmlTe	extWriterA	ttribute						key);			
[C++]	protected:	virtual	void	AddAttribut	te(String*	name,	String*	value,			
HtmlTe	HtmlTextWriterAttribute key);										
[VB] (	Overridable	Protecte	d Sub	AddAttribut	te(ByVal	name As	s String,	ByVal			
value	As	String,	ByVa	al key	As ]	HtmlText	WriterAt	tribute)			
[JScript	t] protecte	d function	n Add	Attribute(nan	ne : Strin	g, value	: String,	key:			
HtmlTe	HtmlTextWriterAttribute);										

Adds the specified HTML attribute and its value, along with an System.Web.UI.HtmlTextWriterAttribute value, to the HtmlTextWriter output stream. The HTML attribute to add. The value to assign to the HTML attribute. An HtmlTextWriterAttribute value that represents the HTML attribute.

# AddStyleAttribute

[C#] public virtual void AddStyleAttribute(HtmlTextWriterStyle key, string value);

[C++] public: virtual void AddStyleAttribute(HtmlTextWriterStyle key, String\* value);

[VB] Overridable Public Sub AddStyleAttribute(ByVal key As HtmlTextWriterStyle, ByVal value As String)

[JScript] public function AddStyleAttribute(key: HtmlTextWriterStyle, value: String);

# Description

Adds the HTML style attribute associated with the System.Web.UI.HtmlTextWriterStyle value specified by the *key* parameter and the attribute's value to the System.Web.UI.HtmlTextWriter output stream. An HtmlTextWriterStyle value that represents the HTML style attribute to add. The value to assign to the HTML attribute.

# AddStyleAttribute

lee@hayes pik 509-324+9256 1094 MS1-863US.APP

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

[C#] public virtual void AddStyleAttribute(string name, string value);
[C++] public: virtual void AddStyleAttribute(String\* name, String\* value);
[VB] Overridable Public Sub AddStyleAttribute(ByVal name As String, ByVal value

As String)

[JScript] public function AddStyleAttribute(name: String, value: String); Adds an HTML style attribute to the to the System.Web.UI.HtmlTextWriter output stream.

## Description

Adds the specified HTML style attribute and its value to the **System.Web.UI.HtmlTextWriter** output stream. The HTML style attribute to add. The value assigned to the HTML style attribute.

# AddStyleAttribute

[C#] protected virtual void AddStyleAttribute(string name, string value, HtmlTextWriterStyle key); [C++] protected: virtual void AddStyleAttribute(String\* name, String\* value, HtmlTextWriterStyle key); [VB] Overridable Protected Sub AddStyleAttribute(ByVal name As String, ByVal value As String, ByVal HtmlTextWriterStyle) key As [JScript] protected function AddStyleAttribute(name : String, value : String, key : HtmlTextWriterStyle);

## Description

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Adds the specified HTML style attribute, along with its value, to the System.Web.UI.HtmlTextWriter output stream. The HTML style attribute to add to the output stream. The value to assign to the HTML attribute. An System.Web.UI.HtmlTextWriterStyle value that represents the HTML style attribute to add.

Close

[C#]	public	override	void	Close();
[C++]	public:		void	Close();
[VB]	Overrides	Public	Sub	Close()
[JScript]	public	override	function	Close();

# Description

Closes the current **System.Web.UI.HtmlTextWriter** and releases any system resources associated with it.

This implementation of Close calls the System.IO.TextWriter method passing a true value.

### EncodeAttributeValue

[C#] protected virtual string EncodeAttributeValue(HtmlTextWriterAttribute value); attrKey, string [C++] protected: virtual String\* EncodeAttributeValue(HtmlTextWriterAttribute attrKey, String\* value); [VB] Overridable Protected Function EncodeAttributeValue(ByVal attrKey As HtmlTextWriterAttribute, String) String ByVal value As As

[JScript] protected function EncodeAttributeValue(attrKey : HtmlTextWriterAttribute, value : String) : String;

Description

3

5

6

7

8

9

10

11

13

14

15

16

17

18

19

20

21

22

23

24

25

value. attribute's HTML specified HTML encodes the value. An encoded attribute The Value: Return System.Web.UI.HtmlTextWriterAttribute value representing the HTML attribute to which the value parameter is assigned. The value assigned to the specified HTML attribute.

EncodeAttributeValue

[C#] protected string EncodeAttributeValue(string value, bool fEncode);
[C++] protected: String\* EncodeAttributeValue(String\* value, bool fEncode);
[VB] Protected Function EncodeAttributeValue(ByVal value As String, ByVal
fEncode As Boolean) As String
[JScript] protected function EncodeAttributeValue(value : String, fEncode :
Boolean) : String; HTML encodes the specified HTML attribute's value.

Description

HTML encodes the specified HTML attribute's value. Return Value: The HTML-encoded attribute value, **null** if the value parameter is empty, or the unencoded attribute value if fEncode is **false**. The attribute value to encode. **true** to HTML encode the attribute value; otherwise, **false**.

EncodeUrl

lee **©**hayes piic 509-324-9255 1097 *MS1-863US.APP* 

Ę
ïLi
Ü
Ľ
H
ا مورود
i da
l ulla

[C#]	protected	string	Encodel	Jrl(string	url);
[C++]	protected:	String*	Encodel	Jrl(String*	url);
[VB] Prot	ected Function	EncodeUrl(ByV	al url As	s String) A	As String
[JScript]	protected fun	ction Encodel	Jrl(url :	String) :	String;
Description					
Perfo	orms minimal Ul	RL encoding by	converting s	paces passed	in the url
parameter		to			"%20".
Return Valu	e: The encoded I	JRL. The URL to	be encoded.		
Filte	rAttributes				
[C#]	protected	virtual	void	FilterA	.ttributes();
[C++]	protected:	virtual	void	FilterA	.ttributes();
[VB]	Overridable	Protected	Sub	Filter	Attributes()
[JScript]	protect	ed fu	nction	FilterA	.ttributes();
Description					
Reno	ders all HTM	L attributes a	and style	attributes b	y calling
System.We	eb.UI.HtmlText	Writer.OnAttrib	uteRender(S	System.String	,System.S
tring,Syste	m.Web.UI.Htm	TextWriterAttri	bute)		and

System.Web.UI.HtmlTextWriter.OnAttributeRender(System.String,System.String,System.Web.UI.HtmlTextWriterAttribute)

and
System.Web.UI.HtmlTextWriter.OnStyleAttributeRender(System.String,System.String,System.String,System.Web.UI.HtmlTextWriterStyle) on all properites of the page or ASP.NET server control.

Flush

[C#]	public	override	void	Flush();
[C++]	public:		void	Flush();
[VB]	Overrides	Public	Sub	Flush()
[JScript]	public	override	function	Flush();

Clears all buffers for the current **System.Web.UI.HtmlTextWriter** and causes any buffered data to be written to the text stream.

### GetAttributeKey

[C#] protected HtmlTextWriterAttribute GetAttributeKey(string attrName);
[C++] protected: HtmlTextWriterAttribute GetAttributeKey(String\* attrName);
[VB] Protected Function GetAttributeKey(ByVal attrName As String) As
HtmlTextWriterAttribute

[JScript] protected function GetAttributeKey(attrName : String) :
HtmlTextWriterAttribute;

### Description

Obtains the corresponding System.Web.UI.HtmlTextWriterAttribute enumeration value for the specified HTML attribute.

Return Value: The HtmlTextWriterAttribute enumeration value for the specified HTML attribute. The HTML attribute to obtain the HtmlTextWriterAttribute value for.

## GetAttributeName

1	
2	[C#] protected string GetAttributeName(HtmlTextWriterAttribute attrKey);
3	[C++] protected: String* GetAttributeName(HtmlTextWriterAttribute attrKey);
4	[VB] Protected Function GetAttributeName(ByVal attrKey As
5	HtmlTextWriterAttribute) As String
6	[JScript] protected function GetAttributeName(attrKey :
7	HtmlTextWriterAttribute) : String;
8	
9	Description
10	Obtains the name of the HTML attribute associated with the specified
11	System.Web.UI.HtmlTextWriterAttribute value.
12	Return Value: The name of the HTML attribute. The HtmlTextWriterAttribute
13	to obtain the HTML attribute name for.
14	GetStyleKey
15	
16	[C#] protected HtmlTextWriterStyle GetStyleKey(string styleName);
17	[C++] protected: HtmlTextWriterStyle GetStyleKey(String* styleName);
18	[VB] Protected Function GetStyleKey(ByVal styleName As String) As
19	HtmlTextWriterStyle
20	[JScript] protected function GetStyleKey(styleName : String) :
21	HtmlTextWriterStyle;
22	
23	Description
24	Obtains the System.Web.UI.HtmlTextWriterStyle enumeration value for
25	the specified HMTL style.

MS1-863US.APP lee@hayes pilc 509+324+9256

3

4

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Description

Return Value: The HtmlTextWriterStyle value. The HTML style attribute to obtain the HtmlTextWriterStyle value for. GetStyleName GetStyleName(HtmlTextWriterStyle [C#] protected string GetStyleName(HtmlTextWriterStyle String\* [C++]protected: styleKey Function GetStyleName(ByVal [VB] Protected HtmlTextWriterStyle) As [JScript] protected function GetStyleName(styleKey : HtmlTextWriterStyle) : String; Description Obtains the HTML style attribute associated with the specified enumeration System.Web.UI.HtmlTextWriterStyle Return Value: The HTML style attribute. The HtmlTextWriterStyle value to obtain the HTML style attribute for. GetTagKey GetTagKey(string HtmlTextWriterTag [C#] virtual protected [C++] protected: virtual HtmlTextWriterTag GetTagKey(String\* tagName); [VB] Overridable Protected Function GetTagKey(ByVal tagName As String) As **HtmlTextWriterTag** [JScript] protected function GetTagKey(tagName : String) : HtmlTextWriterTag;

styleKey);

styleKey);

As

String

value.

tagName);

1101 MS1-863US.APP lee@hayes plic 509+324+9256

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Obtains the System. Web. UI. Html Text Writer Tag enumeration value HTML element. specified the associated with Return Value: The HtmlTextWriterTag value; if the tagName parameter is not **HtmlTextWriterTag** value, specific with associated a HtmlTextWriterTag.Unknown is returned. The HTML element to obtain the HtmlTextWriterTag value for.

GetTagName

GetTagName(HtmlTextWriterTag virtual string protected [C#] [C++] protected: virtual String\* GetTagName(HtmlTextWriterTag tagKey); Protected Function GetTagName(ByVal tagKey Overridable [VB] String HtmlTextWriterTag) As [JScript] protected function GetTagName(tagKey: HtmlTextWriterTag): String;

### Description

specified **HTML** element associated with the **Obtains** the value. enumeration System.Web.UI.HtmlTextWriterTag Return Value: The HTML element. The HtmlTextWriterTag value to obtain the HTML element for.

### **IsAttributeDefined**

Is Attribute Defined (Html Text Writer Attributekey); [C#] protected bool IsAttributeDefined(HtmlTextWriterAttribute key); [C++]protected: bool IsAttributeDefined(ByVal key As Protected Function [VB] Html Text Writer Attribute)Boolean As

1102 MS1-863US APP lee@hayes plic 509+324+9256

[JScript] protected function IsAttributeDefined(key: HtmlTextWriterAttr	ibute):
Boolean; Determines whether an HTML attribute is recognized as one	of the
System.Web.UI.HtmlTextWriterAttribute	values.

3

4

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Determines whether the specified HTML attribute is recognized.

Return Value: true if the HTML attribute is associated with one of the 
HtmlTextWriterAttribute values; otherwise, false . The 
System.Web.UI.HtmlTextWriterAttribute associated with the HTML attribute.

### **IsAttributeDefined**

[C#]	protected	bool	Is Attribute Defined (Html Text Writer Attribute	key,	out	string
value	e);					

[C++] protected: bool IsAttributeDefined(HtmlTextWriterAttribute key, String\*\* value);

IsAttributeDefined(ByVal key As Function [VB] Protected Boolean String) As HtmlTextWriterAttribute, ByRef value As [JScript] protected function IsAttributeDefined(key: HtmlTextWriterAttribute, String) Boolean; value

## Description

Determines whether the specified HTML attribute is recognized.

Return Value: true if the HTML attribute is associated with one of the 
HtmlTextWriterAttribute values; otherwise, false . The

lee@hayes\_glic 509-324-9256 1103 MS1-863US.APP

1	System.Web.UI.HtmlTextWriterAttribute associated with the HTML attribute.
2	The value assigned to the HTML attribute.
3	IsStyleAttributeDefined
4	
5	[C#] protected bool IsStyleAttributeDefined(HtmlTextWriterStyle key);
6	[C++] protected: bool IsStyleAttributeDefined(HtmlTextWriterStyle key);
7	[VB] Protected Function IsStyleAttributeDefined(ByVal key As
8	HtmlTextWriterStyle) As Boolean
9	[JScript] protected function IsStyleAttributeDefined(key: HtmlTextWriterStyle):
10	Boolean; Determines whether an HTML style attribute is recognized as one of the
11	System.Web.UI.HtmlTextWriterStyle values.
12	
13	Description
14	Determines whether the specified HTML style attribute is recognized.
15	Return Value: true if the HTML attribute is associated with one of the
16	HtmlTextWriterStyle values; otherwise, false . The
17	System.Web.UI.HtmlTextWriterStyle value associated with the HTML
18	attribute.
19	IsStyleAttributeDefined
20	
21	[C#] protected bool IsStyleAttributeDefined(HtmlTextWriterStyle key, out string
22	value);
23	[C++] protected: bool IsStyleAttributeDefined(HtmlTextWriterStyle key, String**
24	value);
25	[VB] Protected Function IsStyleAttributeDefined(ByVal key As

HtmlTextWriterStyle, ByRef value As String) As Boolean [JScript] protected function IsStyleAttributeDefined(key: HtmlTextWriterStyle, value : String) : Boolean;

Description

Determines whether the specified HTML style attribute is recognized.

Return Value: true if the HTML attribute is associated with one of the HtmlTextWriterStyle values; otherwise, false . The System.Web.UI.HtmlTextWriterStyle value associated with the HTML attribute. The value assigned to the HTML style attribute.

OnAttributeRender

[C#] protected virtual bool OnAttributeRender(string name, string value,

HtmlTextWriterAttribute key);

[C++] protected: virtual bool OnAttributeRender(String\* name, String\* value,

HtmlTextWriterAttribute key);

[VB] Overridable Protected Function OnAttributeRender(ByVal name As String,

ByVal value As String, ByVal key As HtmlTextWriterAttribute) As Boolean

[JScript] protected function OnAttributeRender(name: String, value: String, key:

HtmlTextWriterAttribute) : Boolean;

22 Description

Determines whether the specified HTML attribute and its value have been rendered to the requesting page.

Return Value: true if the attribute has been rendered to the page; otherwise, false.

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

The HTML attribute to render. The value that is assinged to the HTML attribute.

The System.Web.UI.HtmlTextWriterAttribute enumeration value associated with the HTML attribute.

## OnStyleAttributeRender

### Description

Determines whether the specified HTML style attribute and its value have rendered the requesting been to page. Return Value: true if the HTML style attribute has been rendered to the page; otherwise, false. The HTML style attribute to render. The value that is assinged to attribute. The System.Web.UI.HtmlTextWriterStyle HTML style the enumeration value associated with the HTML style attribute.

## OnTagRender

[C#] protected virtual bool OnTagRender(string name, HtmlTextWriterTag key); [C++] protected: virtual bool OnTagRender(String\* name, HtmlTextWriterTag

[VB] Overridable Protected Function OnTagRender(ByVal name As String, ByVal key As HtmlTextWriterTag) As Boolean [JScript] protected function OnTagRender(name : String, key : HtmlTextWriterTag) : Boolean;

## Description

key);

Determines whether the specified HTML element has been rendered to the requesting page.

\*Return Value: true if if the HTML element has been rendered to the page; otherwise, false . The HTML element to render. The \*System.Web.UI.HtmlTextWriterTag\* enumeration value associated with the HTML element.

# OutputTabs

[C#]	protected	virtual	void	OutputTabs();
[C++]	protected:	virtual	void	OutputTabs();
[VB]	Overridable	Protected	Sub	OutputTabs()
[JScript]	protected	fur	nction	OutputTabs();

### Description

Writes a series of blank characters that represent the tab spacing for a line of HTML characters.

lee@haves oic 509-324-9256 1107 MS1-863US.APP

This method uses the integer passed to the System.Web.UI.HtmlTextWriter.Indent property to determine how many spaces to write for the tab spacing.

## PopEndTag

[C#]	protected		string	Pop	EndTag();
[C++]	protec	eted:	String*	Pop	EndTag();
[VB]	Protected	Function	PopEndTag()	As	String
[JScript]	protected	function	PopEndTag()	:	String;

## Description

Retrieves the appropriate closing tag for the HTML element to render to the page.

Return Value: The appropriate closing tag for the HTML element.

## PushEndTag

[C#]	protected	void	PushEndTag	g(string		endTag);
[C++]	protected:	void	PushEndTag	(String*		endTag);
[VB]	Protected Su	ıb PushEn	dTag(ByVal e	ndTag	As	String)
[JScript]	protected	function	PushEndTag(er	ndTag	:	String);

## Description

Associates the specifed closing tag of an HTML element with the approprite HTML text. The closing tag to be associated.

# RegisterAttribute

1	
2	[C#] protected static void RegisterAttribute(string name, HtmlTextWriterAttribute
3	key);
4	[C++] protected: static void RegisterAttribute(String* name,
5	HtmlTextWriterAttribute key);
6	[VB] Protected Shared Sub RegisterAttribute(ByVal name As String, ByVal key
7	As HtmlTextWriterAttribute)
8	[JScript] protected static function RegisterAttribute(name : String, key :
9	HtmlTextWriterAttribute);
10	
11	Description
12	Registers HTML attributes, whether literals or dynamically generated, from
13	the source file so that they can be properly rendered to the requesting client. The
14	HTML attribute to be registered. An System.Web.UI.HtmlTextWriterAttribute
15	value that corresponds to the attribute name.
16	RegisterStyle
17	
18	[C#] protected static void RegisterStyle(string name, HtmlTextWriterStyle key);
19	[C++] protected: static void RegisterStyle(String* name, HtmlTextWriterStyle
20	key);
21	[VB] Protected Shared Sub RegisterStyle(ByVal name As String, ByVal key As
22	HtmlTextWriterStyle)
23	[JScript] protected static function RegisterStyle(name : String, key :
24	HtmlTextWriterStyle);
25	

Registers HTML style properties, whether literals or dynamically generated, from the source file so that they can be properly rendered to the requesting client. The **System.String** passed from the source file specifying the style name. The **System.Web.UI.HtmlTextWriterStyle** value that corresponds to the specified style.

RegisterTag

[C#] protected static void RegisterTag(string name, HtmlTextWriterTag key);
[C++] protected: static void RegisterTag(String\* name, HtmlTextWriterTag key);
[VB] Protected Shared Sub RegisterTag(ByVal name As String, ByVal key As HtmlTextWriterTag)

[JScript] protected static function RegisterTag(name : String, key : HtmlTextWriterTag);

## Description

Registers HTML tags, whether literals or dynamically generated, from the source file so that they can be properly rendered to the requesting client. A System.String that contains the HTML tag. An System.Web.UI.HtmlTextWriterTag value that specifies which element is to be rendered.

### RenderAfterContent

[C#] protected virtual string RenderAfterContent();

[C++] protected: virtual String\* RenderAfterContent(); [VB] Overridable Protected Function RenderAfterContent() As String [JScript] protected function RenderAfterContent() : String;

### Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Writes any text or spacing that occurs after the content of the HTML element to the **System.Web.UI.HtmlTextWriter** output stream. *Return Value:* The spacing or text to write prior to the content of the HTML element. If not overridden, this method returns **null**.

### RenderAfterTag

[C#] protected virtual string RenderAfterTag(); [C++]protected: virtual String\* RenderAfterTag(); [VB] Overridable Protected Function RenderAfterTag() As String [JScript] RenderAfterTag() protected function String;

## Description

Writes any spacing or text that occurs after an HTML element's closing tag. Return Value: The spacing or text to write after the closing tag of the HTML element. If not overridden, this method returns null.

### RenderBeforeContent

[C#] protected virtual string RenderBeforeContent();
[C++] protected: virtual String\* RenderBeforeContent();
[VB] Overridable Protected Function RenderBeforeContent() As String

[JScript] protected function RenderBeforeContent() : String;

Description

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Writes any tab spacing before the content contained in an HTML element. Return Value: The spacing to write prior to the content of the HTML element. If not overridden, this method returns null.

## RenderBeforeTag

[C#] protected virtual string RenderBeforeTag(); [C++]protected: virtual String\* RenderBeforeTag(); [VB] Overridable Protected Function RenderBeforeTag() As String [JScript] protected function RenderBeforeTag() String;

## Description

Writes any text or tab spacing that occurs before the opening tag of an HTML element to the **System.Web.UI.HtmlTextWriter** output stream. Return Value: The text or tab spacing to write to the output stream. If not overridden, this method returns **null**.

# RenderBeginTag

[C#] public virtual void RenderBeginTag(HtmlTextWriterTag tagKey); [C++]public: virtual void RenderBeginTag(HtmlTextWriterTag [VB] Overridable Public Sub RenderBeginTag(ByVal tagKey As HtmlTextWriterTag)

[JScript] public function RenderBeginTag(tagKey : HtmlTextWriterTag);

Writes the opening tag of the HTML element associated with the specified **System.Web.UI.HtmlTextWriterTag** enumeration value to the output stream. An **HtmlTextWriterTag** value that defines the opening tag of the HTML element to render.

### RenderBeginTag

[C#] public virtual void RenderBeginTag(string tagName); [C++] public: virtual void RenderBeginTag(String\* tagName); [VB] Overridable Public Sub RenderBeginTag(ByVal tagName As String) [JScript] public function RenderBeginTag(tagName : String); Writes the opening tag of an HTML element to the System.Web.UI.HtmlTextWriter output stream.

## Description

Writes the opening tag of the specified HTML element to the output stream. The HTML element to render the opening tag for.

## RenderEndTag

[C#]	public	virtual	void	RenderEndTag();
[C++]	public:	virtual	void	RenderEndTag();
[VB]	Overridable	Public	Sub	RenderEndTag()
[JScript]	public	fur	nction	RenderEndTag();

# Description

.:	ısı.
.;	1
1,	
i,	Total
í	4
- 12	3.5
1	ij.
Į,	ij
!	1 477 111
1	
::	
i,	
٤,	ı, İ
ľ	
,;	inite.
ł,	4
;	alla.

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Writes the end tag of an HTML element to the System.Web.UI.HtmlTextWriter output stream.

Write

[C#] public override void Write(bool value); [C++]public: void Write(bool value); [VB] Overrides Sub Public Write(ByVal value As Boolean) [JScript] public override function Write(value Boolean);

# Description

Writes the text representation of a **Boolean** value to the text stream, along with any specified tab spacing. The **Boolean** value to be written to the text stream.

Write

[C#] public override void Write(char value); [C++]public: void Write( wchar t value); [VB] Overrides Public Sub Write(ByVal value As Char) [JScript] public override function Write(value Char);

# Description

Writes a unicode character to the text stream, along with any specified tab spacing. The unicode character to write to the text stream.

Write

[C#] public override void Write(char[] buffer);

lee@hayes ρlic 509+324+9256 11114

MS1-863US.APP

[C++]public: void Write( wchar t buffer \_gc[]); Overrides [VB] Public Sub Write(ByVal buffer() Char) As [JScript] public override function Write(buffer Char[]);

## Description

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Writes a character array to the text stream, along with any specified tab spacing. The character array to write to the text stream.

Write

[C#] public override void Write(double value): [C++]public: void Write(double value); [VB] Overrides Public Sub Write(ByVal value Double) As [JScript] public override function Write(value double);

# Description

Writes the text representation of a double-precision floating point number to the HTML text stream, along with any specified tab spacing. The double-precision floating point number to write to the text stream.

Write

[C#] public override void Write(int value); [C++]public: void Write(int value); [VB] Overrides Public Sub Write(ByVal value Integer) As [JScript] public override function Write(value int);

Writes the text representation of a 32-byte signed integer to the text stream, along with any specified tab spacing. The 32-byte signed integer to write to the text stream.

Write

[C#]	public	overr	ide	void	V	Vrite(long		value);
[C++]	publ	ic:	voic	ł	Write(_	int64		value);
[VB]	Overrides	Public	Sub	Write(I	ByVal	value	As	Long)
[JScript]	public	overrid	e	function	Writ	e(value	:	long);

# Description

Writes the text representation of an 64-byte signed integer to the text stream, along with any specified tab spacing. The 64-byte signed integer to write to the text stream.

Write

[C#]	public	overr	ide	void	W	/rite(obj	ect	value);
[C++]	publ	ic:	void		Write(	Object*		value);
[VB]	Overrides	Public	Sub	Write(1	ByVal	value	As	Object)
[JScript]	public	overrid	e fu	inction	Write	e(value	:	Object);

# Description

lee@hayes pilc 509-324-9256 11116 MS1-863US,APP

;	100 kg
	1
١,	Ľ
į	351 100 100 100 100 100 100 100 100 100 1
ľ	L. L. K. 3.11
i,	ij
ť,	## ##
:	
ij	
i,	
	4
	÷
ľ	
i,	:31g
	=la

Writes the text representation of an **System.Object** to the text stream, along with any specified tab spacing. The **Object** to write to the text stream.

Write

1 |1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] public override void Write(float value); [C++]public: void Write(float value); [VB] Overrides Public Sub Write(ByVal value As Single) [JScript] public override function Write(value float);

## Description

Writes the text representation of a single-precision floating point number to the HTML text stream, along with any specified tab spacing. The single-precision floating point number to write to the text stream.

Write

[C#] public override void Write(string s); [C++]public: void Write(String\* s); [VB] **Overrides** Public Sub Write(ByVal As String) [JScript] public override function Write(s : String); Writes the given data type to HTML an text stream to be rendered to ASP.NET an page.

### Description

Writes the specified string to the text stream, along with any specified tab spacing. A **System.String** to be written to the text stream.

Write

6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

22

23

24

2

[C#] public override void Write(string format, object arg0); Write(String\* Object\* [C++]public: void format, arg0); [VB] Overrides Public Sub Write(ByVal format As String, ByVal arg0 As Object) [JScript] public override function Write(format : String, arg0 : Object);

### Description

Writes a tab string and a formatted string to the HTML text stream, using the same semantics as System.String.Format(System.String,System.Object). The formatting string. An object to write into the formatted string.

Write

public override void Write(string format, params object[] [C#] arg); [C++]public: void Write(String\* format, Object\* gc[]); arg [VB] Overrides Public Sub Write(ByVal format As String, ByVal ParamArray As Object) arg() [JScript] public override function Write(format : String, arg : Object[]);

## Description

Writes a tab string and a formatted string to the HTML text stream, using the same semantics as System.String.Format(System.String,System.Object). The formatting string. The object array to write into the formatted string.

Write

public override void Write(char[] buffer, int index, int count);

1118 lee@hayes pilc 509-324-9256 MS1-863US.APF [C++] public: void Write(\_wchar\_t buffer \_gc[], int index, int count);
[VB] Overrides Public Sub Write(ByVal buffer() As Char, ByVal index As
Integer, ByVal count As Integer)
[JScript] public override function Write(buffer : Char[], index : int, count : int);

## Description

Writes a subarray of characters to the text stream, along with any specified tab spacing. The array of characters from which the subarray is written to the text stream. The index location in the array where writing begins. The number of characters to be written to the text stream.

Write

[C#] public override void Write(string format, object arg0, object arg1);
[C++] public: void Write(String\* format, Object\* arg0, Object\* arg1);
[VB] Overrides Public Sub Write(ByVal format As String, ByVal arg0 As Object,
ByVal arg1 As Object)
[JScript] public override function Write(format : String, arg0 : Object, arg1 : Object);

## Description

Writes a tab string and a formatted string to the HTML text stream, using the same semantics as **System.String.Format(System.String,System.Object)**. The formatting string. An object to write into the formatted string. An object to write into the formatted string.

WriteAttribute

15

17

18

19

20

21

22

23

24

1

3

[C#] public virtual void WriteAttribute(string name, string value); [C++] public: virtual void WriteAttribute(String\* name, String\* value); [VB] Overridable Public Sub WriteAttribute(ByVal name As String, ByVal value As String) [JScript] public function WriteAttribute(name : String, value : String); Writes an HTML attribute and its value to the System.Web.UI.HtmlTextWriter output stream.

### Description

Writes the specified HTML attribute and value to the **System.Web.UI.HtmlTextWriter** output stream. The HTML attribute to write to the output stream. The value assigned to the HTML attribute.

### WriteAttribute

[C#] public virtual void WriteAttribute(string name, string value, bool fEncode); [C++] public: virtual void WriteAttribute(String\* name, String\* value, bool fEncode);

[VB] Overridable Public Sub WriteAttribute(ByVal name As String, ByVal value As String, ByVal fEncode As Boolean)

[JScript] public function WriteAttribute(name : String, value : String, fEncode : Boolean);

### Description

25

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Writes the specified HTML attribute and value to the **System.Web.UI.HtmlTextWriter** output stream and HTML encodes them if specified in the *fEncode* parameter. The HTML attribute to write to the output stream. The value assigned to the HTML attribute. **true** to HTML encode the attribute and its assigned value; otherwise, **false**.

## WriteBeginTag

[C#] public virtual void tagName); WriteBeginTag(string [C++]public: virtual void WriteBeginTag(String\* tagName); [VB] Overridable Public Sub WriteBeginTag(ByVal tagName As String) [JScript] public function WriteBeginTag(tagName String);

## Description

Writes any tab spacing and the opening tag of the specified HTML element to the **System.Web.UI.HtmlTextWriter** output stream.

This method does not write the closing character (>) of the HTML element's opening tag. Use this method with the **System.Web.UI.HtmlTextWriter.SelfClosingTagEnd** constant when you write HTML elements that are self closing. The HTML element to write the opening tag of.

# WriteEndTag

[C#] public virtual void WriteEndTag(string tagName); [C++] public: virtual void WriteEndTag(String\* tagName); [VB] Overridable Public Sub WriteEndTag(ByVal tagName As String) [JScript] public function WriteEndTag(tagName : String);

Description

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Writes any tab spacing and the closing tag of the specified HTML element.

The HTML element to write the closing tag for.

WriteFullBeginTag

[C#] public virtual void WriteFullBeginTag(string tagName); [C++]public: virtual void WriteFullBeginTag(String\* tagName); [VB] Overridable Public Sub WriteFullBeginTag(ByVal tagName As String) [JScript] public function WriteFullBeginTag(tagName String);

## Description

Writes any tab spacing and the opening tag of the specified HTML element to the **System.Web.UI.HtmlTextWriter** output stream.

This method automatically writes the closing character (>) of the opening tag of the HTML element. The HTML element to write.

WriteLine

[C#]	public	override	void	WriteLine();
[C++]	public:		void	WriteLine();
[VB]	Overrides	Public	Sub	WriteLine()
[JScript]	public	override	function	WriteLine();

Description

Writes a line terminator to the HTML text stream.
WriteLine

[C#] public WriteLine(bool override void value); [C++]public: void WriteLine(bool value); [VB] Overrides Public Sub WriteLine(ByVal value Boolean) [JScript] public override function WriteLine(value Boolean);

Description

1

2

3

6

7

8

9

10

11

13

14

15

16

17

18

19

20

21

22

23

24

Writes a tab string and the text representation of a **Boolean** followed by a line terminator to the HTML text stream. The **Boolean** to be written to the text stream.

WriteLine

[C#] public override void WriteLine(char value); [C++]public: void WriteLine( wchar t value); [VB] Overrides Public Sub WriteLine(ByVal value As Char) [JScript] public override function WriteLine(value Char);

Description

Writes a tab string and a character followed by a line terminator to the HTML text stream. The character to be written to the text stream.

WriteLine

[C#] public override void WriteLine(char[] buffer);

[C++] public: void WriteLine( wchar t buffer \_gc[]); Overrides Public [VB] Sub WriteLine(ByVal buffer() Char) [JScript] public override function WriteLine(buffer Char[]);

## Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Writes a tab string and a character array followed by a line terminator to the HTML text stream. The character array to be written to the text stream.

### WriteLine

[C#] public override void WriteLine(double value); [C++]public: void WriteLine(double value); Overrides [VB] **Public** Sub WriteLine(ByVal value As Double) [JScript] public override function WriteLine(value double);

## Description

Writes a tab string and the text representation of an 8-byte floating-point value, followed by a line terminator, to the HTML text stream. The 8-byte floating-point value to write to the text stream.

### WriteLine

[C#] public override void WriteLine(int value); [C++]public: void WriteLine(int value); [VB] Overrides WriteLine(ByVal Public Sub value Integer) As [JScript] public override function WriteLine(value int);

lee@hayes pilc 509-324-9256 1124 MS1-863US.APP

Writes a tab string and the text representation of an 32-byte signed integer, followed by a line terminator, to the HTML text stream. The 32-byte signed integer to write to the text stream.

## WriteLine

[C#]	public	overn	ride	void	WriteI	Line(long		value);
[C++]	publi	ic:	void		WriteLine(_	_int64		value);
[VB]	Overrides	Public	Sub	WriteL	ine(ByVal	value	As	Long)
[JScript	public	overrio	de f	unction	WriteLin	e(value	:	long);

## Description

Writes a tab string and the text representation of an 64-byte signed integer, followed by a line terminator, to the HTML text stream. The 64-byte signed integer to write to the text stream.

### WriteLine

[C#]	public	overr	ide	void	WriteI	Line(obje	ect	value);
[C++]	publ	ic:	void		WriteLine(	Object*		value);
[VB]	Overrides	Public	Sub	WriteL	ine(ByVal	value	As	Object)
[JScript	] public	overrid	le fi	unction	WriteLin	e(value	:	Object);

## Description

lee@hayes pilc 509-324-9256

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Writes any tab strings and the text representation of an **System.Object**, followed by a line terminator, to the HTML text stream. The **Object** to write to the text stream.

WriteLine

[C#] public override void WriteLine(float value); public: WriteLine(float [C++]void value); Overrides Sub [VB] Public WriteLine(ByVal value Single) As [JScript] public override function WriteLine(value float);

### Description

Writes a tab string and the text representation of a single-precision floating point number, followed by a line terminator, to the HTML text stream. The single-precision floating point number to write to the text string.

WriteLine

override void [C#] public WriteLine(string s); void [C++]public: WriteLine(String\* s); [VB] Overrides Public Sub WriteLine(ByVal As String) [JScript] public override function WriteLine(s : String); Writes some data to an HTML text stream as specified by the overloaded parameters, followed by a line terminator. All instances of this method write tab strings to the text stream.

Description

 Writes a tab string and a **System.String** followed by a line terminator to an HTML text stream. The **String** to write to the text stream.

WriteLine

1

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] public override void WriteLine(uint value); [C++]public: void WriteLine(unsigned int value); [VB] Overrides Sub Public WriteLine(ByVal value As UInt32) [JScript] public override function WriteLine(value UInt32);

## Description

Writes any tab strings and the text representation of a 4-byte unsigned integer followed by a line terminator to the HTML text stream. The 4-byte unsigned integer to write.

### WriteLine

public [C#] override void WriteLine(string format, object arg0); [C++]public: void WriteLine(String\* format, Object\* arg0); [VB] Overrides Public Sub WriteLine(ByVal format As String, ByVal arg0 As Object)

[JScript] public override function WriteLine(format : String, arg0 : Object);

### Description

Writes any tab strings and a formatted string, followed by a line terminator, to the HTML text stream. The method uses the same semantics as

lee@hayes plic 509-324-9256

**System.String.Format(System.String,System.Object)** . The formatting string. The object to write into the formatted string.

WriteLine

[C#] public override void WriteLine(string format, params object[] arg);
[C++] public: void WriteLine(String\* format, Object\* arg \_\_gc[]);
[VB] Overrides Public Sub WriteLine(ByVal format As String, ByVal
ParamArray arg() As Object)
[JScript] public override function WriteLine(format : String, arg : Object[]);

## Description

Writes any tab strings and a formatted string, followed by a line terminator, to the HTML text stream. The method uses the same semantics as System.String.Format(System.String,System.Object). The formatting string. The array of objects to write into the formatted string.

WriteLine

[C#] public override void WriteLine(char[] buffer, int index, int count);
[C++] public: void WriteLine(\_wchar\_t buffer \_gc[], int index, int count);
[VB] Overrides Public Sub WriteLine(ByVal buffer() As Char, ByVal index As
Integer, ByVal count As Integer)
[JScript] public override function WriteLine(buffer : Char[], index : int, count : int);

Description

Writes a tab string and a subarray of characters followed by a line terminator to the HTML text stream. The character array from which to write to the text stream. The location in the character array where writing begins. The number of characters in the array to write to the text stream.

WriteLine

6

1

2

3

4

5

7

9

11 12

13 14

15 16

18

17

19 20

21

23

24

25

[C#] public override void WriteLine(string format, object arg0, object arg1);
[C++] public: void WriteLine(String\* format, Object\* arg0, Object\* arg1);
[VB] Overrides Public Sub WriteLine(ByVal format As String, ByVal arg0 As Object, ByVal arg1 As Object)
[JScript] public override function WriteLine(format: String, arg0: Object, arg1: Object);

## Description

Writes any tab strings and a formatted string, followed by a line terminator, to the HTML text stream. The method uses the same semantics as **System.String.Format(System.String,System.Object)**. The formatting string. An object to write into the formatted string. An object to write into the formatted string.

### WriteLineNoTabs

[C#] public void WriteLineNoTabs(string s); [C++]public: void WriteLineNoTabs(String\* s); [VB] Public Sub WriteLineNoTabs(ByVal As String) [JScript] public function WriteLineNoTabs(s String);

Writes a **System.String** followed by a line terminator to an HTML text stream. This method ignores any specified tab spacing. The **String** to write to the HTML text stream.

WriteStyleAttribute

[C#] public virtual void WriteStyleAttribute(string name, string value);
[C++] public: virtual void WriteStyleAttribute(String\* name, String\* value);
[VB] Overridable Public Sub WriteStyleAttribute(ByVal name As String, ByVal value

As String)
[JScript] public function WriteStyleAttribute(name: String, value: String); Writes an HTML style attribute and its value to the System.Web.UI.HtmlTextWriter output

## Description

Writes the specified HTML style attribute to the **System.Web.UI.HtmlTextWriter** ouput stream. The HTML style attribute to write to the output stream. The value assigned to the HTML style attribute.

# WriteStyleAttribute

[C#] public virtual void WriteStyleAttribute(string name, string value, bool fEncode);

[C++] public: virtual void WriteStyleAttribute(String\* name, String\* value, bool fEncode);

[VB] Overridable Public Sub WriteStyleAttribute(ByVal name As String, ByVal value As String, ByVal fEncode As Boolean)

[JScript] public function WriteStyleAttribute(name : String, value : String, fEncode : Boolean);

## Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Writes the specified HTML attribute and value to the **System.Web.UI.HtmlTextWriter** output stream and HTML encodes them if specified in the *fEncode* parameter. The HTML style attribute to write to the output stream. The value assigned to the HTML style attribute. **true** to HTML encode the style attribute and its assigned value; otherwise, **false**.

HtmlTextWriterAttribute enumeration (System.Web.UI) WriteStyleAttribute

## Description

Specifies the HTML attributes that an System.Web.UI.HtmlTextWriter or System.Web.UI.Html32TextWriter object writes to the opening tag of an HTML element when a Web request is processed.

# WriteStyleAttribute

[C#] public **HtmlTextWriterAttribute** const Accesskey; [C++]public: const HtmlTextWriterAttribute Accesskey; [VB] Public Const Accesskey As HtmlTextWriterAttribute [JScript] public Accesskey var HtmlTextWriterAttribute;

1							
2	Descripti	on					
3	Sp	ecifies that th	ne HTML acc	esskey a	ttribute sho	uld be writter	n to the tag.
4		riteStyleAttril					
5							
6	[C#]	public	const	F	ItmlTextW	riterAttribute	Align;
7	[C++]	public:	const	]	HtmlTextW	riterAttribute	e Align;
8	[VB]	Public	Const	Align	As	HtmlText	WriterAttribute
9	[JScript]	public	var	Align	:	HtmlTextV	VriterAttribute;
10							
11	Descriptio	on					
12	Spe	ecifies that the	e HTML <b>ali</b> g	g <b>n</b> attribi	ite should l	e written to t	he tag.
13	Wr	iteStyleAttrib	oute				
14							
15	[C#]	public	const		HtmlTextV	VriterAttribut	e Alt;
16	[C++]	public:	const	<del>,</del>	HtmlText	WriterAttribu	te Alt;
17	[VB]	Public	Const	Alt	As	HtmlTextV	VriterAttribute
18	[JScript]	public	var	Alt	:	HtmlTextW	riterAttribute;
19	ъ						
20	Description						
21				attribute	should be v	vritten to the	tag.
22	Wn	teStyleAttrib	ute				
23	[C#]	11'					
24	[C#]	public	const		extWriterA		Background;
25	[C++]	public:	const	HtmlT	extWriterA	ttribute	Background;

1	[VB]	Public	Const	Background	As	HtmlTextWri	terAttribute
2	[JScript]	public	var	Background	:	HtmlTextWrite	erAttribute;
3							
4	Descripti	on					
5	Sp	ecifies that	the HTML	background a	ttribute	should be written	to the tag.
6	$\mathbf{W}_{1}$	riteStyleAtt	ribute				
7							
8	[C#]	public	cons	st HtmlT	extWrit	erAttribute	Bgcolor;
9	[C++]	public:	cor	nst Html7	TextWri	terAttribute	Bgcolor;
10	[VB]	Public	Const	Bgcolor	As	HtmlTextWrit	erAttribute
11	[JScript]	public	var	Bgcolor	:	HtmlTextWrite	erAttribute;
12							
13	Descriptio						
14				bgcolor attribu	te shoul	d be written to th	e tag.
15	Wr	riteStyleAttr	ibute				
16	FOUR						
17	[C#]	public	cons	st Html7	CextWri	terAttribute	Border;
18	[C++]	public:	cor	nst Html	ΓextWri	terAttribute	Border;
19	[VB]	Public	Const	Border	As	HtmlTextWrite	erAttribute
20	[JScript]	public	var	Border	:	HtmlTextWrite	rAttribute;
21	70						
22	Descriptio						
23				<b>border</b> attribute	should	be written to the	tag.
24	Wri	iteStyleAttri	bute				
25							

i .						
[C#]	public	const	HtmlTex	tWriterA	Attribute	Bordercolor;
[C++]	public:	cons	t HtmlTex	ktWriter/	Attribute	Bordercolor;
[VB]	Public	Const	Bordercolor	As	HtmlText	WriterAttribute
[JScript]	public	var	Bordercolor	:	HtmlText	WriterAttribute;
Descripti	on					
Sp	ecifies that	the HTML	bordercolor a	ttribute s	hould be wr	itten to the tag.
W:	riteStyleAt	tribute				
[C#]	public	const	HtmlText	WriterA	ttribute	Cellpadding;
[C++]	public:	const	HtmlTex	tWriterA	Attribute	Cellpadding;
[VB]	Public	Const	Cellpadding	As	HtmlText	WriterAttribute
[JScript]	public	var	Cellpadding	:	HtmlTextV	WriterAttribute;
Description						
			cellpadding att	ribute sh	ould be wri	tten to the tag.
Wr	riteStyleAtt	ribute				
F. G. (7)						
[C#]	public	const	HtmlText	WriterA	ttribute	Cellspacing;
[C++]	public:	const	HtmlTex	tWriterA	ttribute	Cellspacing;
[VB]	Public	Const	Cellspacing	As	HtmlText	WriterAttribute
[JScript]	public	var	Cellspacing	:	HtmlTextW	VriterAttribute;
Descriptio	on					

	1	l
	2	
	3	
	4	
	5	
	6	
	7	
	8	-
	9	
	10	
	11	
	12	
- 15 - 25 - 25	13	
: <b></b>	14	
	15	
	16	
	17	
	18	
	19	
	20	
	21	
	22	
	23	
	24	

Specifies that the HTML **cellspacing** attribute should be written to the tag. WriteStyleAttribute

[C#]	public	const	Html'	TextWrite	erAttribute	Checked;
[C++]	public:	const	Html	TextWrite	erAttribute	Checked;
[VB]	Public	Const	Checked	As	HtmlTextW	riterAttribute
[JScript]	public	var	Checked	:	HtmlTextWr	iterAttribute;

### Description

Specifies that the HTML **checked** attribute should be written to the tag. WriteStyleAttribute

[C#]	public	const		HtmlTextW	riterAttribute	Class;
[C++]	public:	const		HtmlTextW	riterAttribute	Class;
[VB]	Public	Const	Class	As	HtmlTextWriterA	ttribute
[JScript]	public	var	Class	s :	HtmlTextWriterAt	tribute;

#### Description

Specifies that the HTML **class** attribute should be written to the tag. WriteStyleAttribute

[C#]	public	const		HtmlTextV	WriterAttribute	Cols;
[C++]	public:	const	Ė	HtmlText	WriterAttribute	Cols;
[VB]	Public	Const	Cols	As	HtmlTextWriterA	Attribute
[JScript]	public	var	Cols	:	HtmlTextWriterA	ttribute:

1
ļ
ı. 🗖
ľŌ
ij
H
' "į
ı alı
]; = <b>]</b> L

1							
2	Descripti	on					
3	Sp	ecifies that t	he HTML <b>c</b>	ols attribute s	should b	e written to the t	ag.
4	W:	riteStyleAttri	ibute				
5							
6	[C#]	public	const	Html	ΓextWrit	terAttribute	Colspan;
7	[C++]	public:	const	Html	TextWri	terAttribute	Colspan;
8	[VB]	Public	Const	Colspan	As	HtmlTextWr	iterAttribute
9	[JScript]	public	var	Colspan	:	HtmlTextWri	terAttribute;
10							
11	Description	on					
12	Sp	ecifies that the	he HTML <b>c</b>	<b>olspan</b> attrib	ute shou	ld be written to t	he tag.
13	$\mathbf{W}_{1}$	riteStyleAttri	bute				
14						•	
15	[C#]	public	const	HtmlT	extWrite	erAttribute	Disabled;
16	[C++]	public:	const	Html	TextWrit	terAttribute	Disabled;
17	[VB]	Public	Const	Disabled	As	HtmlTextWri	iterAttribute
18	[JScript]	public	var	Disabled	:	HtmlTextWrit	terAttribute;
19							
20	Description	on					
21	Sp	ecifies that the	he HTML <b>d</b>	<b>isabled</b> attrib	oute shou	ald be written to	the tag.
22	Wi	riteStyleAttri	bute				
23							
24	[C#]	public	cons	st H1	mlTextV	WriterAttribute	For;
25	[C++]	public:	cor	nst H	tmlText	WriterAttribute	For;

	1
	2
	3
	4
	5
	6
	7
	8
	9
1122	10
Hart from Said State State	11
113	12
	13
В	14
	15
	16
i di	17
	18
	19
	20
	21
	22
	23
	- 11

[VB]	Public	Const	For	As	HtmlTextWriterAttribute
[JScript]	public	var	For	:	HtmlTextWriterAttribute;
			<b>r</b> attribute	should b	e written to the tag.
[C#]	public	const	Ht	mlTextW	riterAttribute Height;
[C++]	public:	const	H	mlTextW	riterAttribute Height;
[VB]	Public	Const	Height	As	HtmlTextWriterAttribute
[JScript]	public	var	Height	:	HtmlTextWriterAttribute;
	•		<b>ight</b> attrib	ute shoule	d be written to the tag.
[C#]	public	const	Н	tmlTextV	VriterAttribute Href;
[C++]	public:	const	E	ItmlText\	VriterAttribute Href;
[VB]	Public	Const	Href	As	HtmlTextWriterAttribute
[JScript]	public	var	Href	:	HtmlTextWriterAttribute;
			of attribute	should b	e written to the tag.

ľ	17.7
ľ	IJ
:	Ü
Ϊ,	
:	
::	
:	3
;	14
	uli.
ľ	3
:	alla

1							
2	[C#]	public	co	onst	HtmlTex	ktWriterAttribut	e Id;
3	[C++]	public	c: const		HtmlTe	te Id;	
4	[VB]	Public	Const	Id	l As HtmlText		riterAttribute
5	[JScript]	public	var	Id	:	HtmlTextW	riterAttribute;
6							
7	Descripti	on					
8	Sp	ecifies that t	he HTML	id attribute s	should be	written to the ta	ng.
9	W	riteStyleAttr	ibute				
10							
11	[C#]	public	const	Html7	TextWrite	rAttribute	Maxlength;
12	[C++]	public:	const	Html'	TextWrite	erAttribute	Maxlength;
13	[VB]	Public	Const	Maxlength	As	HtmlTextW	riterAttribute
14	[JScript]	public	var	Maxlength	n :	HtmlTextWi	riterAttribute;
15							
16	Description	on					
17	Sp	ecifies that the	ne HTML ı	maxlength a	ttribute s	hould be writter	n to the tag.
18	Wı	riteStyleAttri	bute				
19							
20	[C#]	public	const	Htm	lTextWrit	terAttribute	Multiple;
21	[C++]	public:	const	t Htm	ılTextWri	terAttribute	Multiple;
22	[VB]	Public	Const	Multiple	As	HtmlTextW	riterAttribute
23	[JScript]	public	var	Multiple	:	HtmlTextWr	iterAttribute;
24							
25	Descriptio	on					

	1	Specifies that the HTML multiple attribute should be written to the tag.										
	2	N	/riteStyleAtt	ribute								
	3											
	4	[C#]	public	cons	t Htr	nlTextW	riterAttribute	Name;				
	5	[C++]	public	con	st Ht	mlTextW	VriterAttribute	Name;				
	6	[VB]	Public	Const	Name	As	HtmlTextW	riterAttribute				
	7	[JScript]	public	var	Name	:	iterAttribute;					
	8											
	9	Descripti	ion									
	10	Sp	pecifies that	the HTML r	name attribut	e should	be written to th	e tag.				
ij	11											
	12											
	[C#] public const HtmlTextWriterAttribute											
H	14	[C++]	public:	const	t Htm	lTextWr	iterAttribute	Nowrap;				
	15	[VB]	Public	Const	Nowrap	As	HtmlTextW1	riterAttribute				
	16	[JScript]	public	var	Nowrap	:	HtmlTextWri	iterAttribute;				
į d	17											
	18	Descripti	on									
	19	Sp	ecifies that t	he HTML n	<b>owrap</b> attrib	ute shou	ld be written to	the tag.				
	20	Wı	riteStyleAttr	ibute								
	21											
	22	[C#]	public	const	HtmlTe	extWrite	Attribute	Onchange;				
	23	[C++]	public:	const	HtmlT	extWrite	rAttribute	Onchange;				
	24	[VB] Public Const Onchange As HtmlTextWriterAttribu										
	25	[JScript]	public	var	Onchange	:	HtmlTextWri	terAttribute;				

lee@hayes plic 509+324+9256

1139

MS1-863US.APP

1										
2	Descript	ion								
3	Sı	pecifies that	the HTML o	nchange atti	ribute sh	ould be writter	n to the tag.			
4	W	riteStyleAttı	ribute							
5										
6	[C#]	public	const	Html	TextWri	terAttribute	Onclick;			
7	7 [C++] public: const HtmlTextWriterAttribute Oncl						Onclick;			
8	[VB]	Public	Const	Onclick	As	HtmlTextV	VriterAttribute			
9	[JScript]	public	var	Onclick	:	HtmlTextW	riterAttribute;			
10										
11	Descripti	on								
12	Specifies that the HTML <b>onclick</b> attribute should be written to the tag.									
13	W:	riteStyleAttr	ibute							
14										
15	[C#]	public	const	HtmlTe	extWrite	rAttribute	ReadOnly;			
16	[C++]	public:	const	HtmlT	extWrite	rAttribute	ReadOnly;			
17	[VB]	Public	Const	ReadOnly	As	HtmlTextW	riterAttribute			
18	[JScript]	public	var	ReadOnly	:	HtmlTextWi	riterAttribute;			
19	<b>.</b>									
20	Descriptio		-							
21				adonly attrib	oute shou	ıld be written t	to the tag.			
22	Wr	riteStyleAttri	bute							
23	[C#]				<b>.</b>					
24	[C#]	public	const			iterAttribute	Rows;			
25		public:	const	Htm	ıl TextWı	riterAttribute	Rows;			

	1	[VB]	Public	Const	Rows	As	HtmlTextWi	riterAttribute						
	2	[JScript]	public	var	Rows	:	HtmlTextWr	iterAttribute;						
	3													
	4	Descripti	Description											
	5	Sp	ecifies that the	he HTML r	ows attribut	e should	be written to the	tag.						
	6	W	riteStyleAttri	bute										
	7													
	8	[C#]	public	const	Html	ΓextWrit	erAttribute	Rowspan;						
	9	[C++]	public:	const	Html	TextWri	terAttribute	Rowspan;						
	10	[VB]	Public	Const	Rowspan	As	HtmlTextWr	riterAttribute						
10 10	11	[JScript]	public	var	Rowspan	:	HtmlTextWri	terAttribute;						
	12													
	13	Description	on											
H	14	Spe	ecifies that the	ne HTML <b>r</b>	owspan attri	ibute sho	ould be written to	the tag.						
	15	WriteStyleAttribute												
	16													
jı allı	17	[C#]	public	const	Htm	nlTextW	riterAttribute	Rules;						
	18	[C++]	public:	cons	st Ht:	mlTextW	riterAttribute	Rules;						
	19	[VB]	Public	Const	Rules	As	HtmlTextWr	iterAttribute						
	20	[JScript]	public	var	Rules	:	HtmlTextWri	terAttribute;						
	21	<b>.</b>												
	22	Description												
	23				ıles attribute	should l	be written to the	tag.						
	24	Wr	iteStyleAttrib	oute										
	25													

]						
[C#]	public	const	Htm	lTextWri	terAttribute	Selected;
[C++]	public:	const	Htm	ılTextWr	iterAttribute	Selected;
[VB]	Public	Const	Selected	As	HtmlTextWrit	erAttribute
[JScript]	public	var	Selected	:	HtmlTextWrite	erAttribute;
Descriptio	n					
Spe	ecifies that th	ne HTML <b>se</b>	lected attri	bute shou	ıld be written to tl	ne tag.
						C
[C#]	public	const	Н	tmlTextV	VriterAttribute	Size;
[C++]	public:	cons	st H	ItmlText\	WriterAttribute	Size;
[VB]	Public	Const	Size	As	HtmlTextWrit	erAttribute
[JScript]	public	var	Size	:	HtmlTextWrite	rAttribute;
Description	n					
Spe	cifies that th	e HTML <b>siz</b>	ze attribute	should be	e written to the tag	ζ.
						•
[C#]	public	const	H	tmlText\	VriterAttribute	Src;
[C++]	public:	cons	st F	ItmlText`	WriterAttribute	Src;
[VB]	Public	Const	Src	As	HtmlTextWrite	rAttribute
[JScript]	public	var	Src	:	HtmlTextWrite	:Attribute:
						· · · · · · · · · · · · · · · · · · ·
	[VB] [JScript]  Description Spectription  [C#] [VB] [JScript]  Description Spectription Spectrip	[C++] public: [VB] Public [JScript] public  Description  Specifies that the WriteStyleAttril  [C#] public [C++] public: [VB] Public [JScript] public  Description  Specifies that the WriteStyleAttril  [C#] public [JScript] public  [JScript] public  [C++] public  Pescription  Specifies that the WriteStyleAttril  [C#] public [C++] public: [VB] Public	[C++] public: const [VB] Public Const [JScript] public var  Description  Specifies that the HTML set WriteStyleAttribute  [C#] public const [C++] public: const [VB] Public Const [JScript] public var  Description  Specifies that the HTML siz WriteStyleAttribute  [C#] public const [C++] public const [JScript] public var	[C++] public: const Htm [VB] Public Const Selected [JScript] public var Selected  Description  Specifies that the HTML selected attri WriteStyleAttribute  [C#] public const H [C++] public: const Size [JScript] public Var Size  [JScript] public var Size  Description  Specifies that the HTML size attribute WriteStyleAttribute  [C#] public const H [C++] public const H [C++] public Const Size  [C#] Public Const H [C++] public Const H [C++] public: const H [VB] Public Const Sre	[C++] public: const HtmlTextWr [VB] Public Const Selected As [JScript] public var Selected :  Description  Specifies that the HTML selected attribute show WriteStyleAttribute  [C#] public const HtmlTextWr [C++] public: const HtmlTextWr [VB] Public Const Size As [JScript] public var Size :  Description  Specifies that the HTML size attribute should be WriteStyleAttribute  [C#] public const HtmlTextWr [C#] public var Size :	[C++] public: const HtmlTextWriterAttribute [VB] Public Const Selected As HtmlTextWrite [JScript] public var Selected : HtmlTextWrite  Description  Specifies that the HTML selected attribute should be written to the WriteStyleAttribute  [C#] public const HtmlTextWriterAttribute [C++] public: const HtmlTextWriterAttribute [VB] Public Const Size As HtmlTextWrite [JScript] public var Size : HtmlTextWrite  Description  Specifies that the HTML size attribute should be written to the tag WriteStyleAttribute  [C#] public const HtmlTextWriterAttribute [C#] public const HtmlTextWriterAttribute [C++] public: const HtmlTextWriterAttribute [VB] Public Const Src As HtmlTextWriterAttribute [VB] Public Const Src As HtmlTextWriterAttribute

	1	Specifies that the HTML <b>src</b> attribute should be written to the tag.											
	2	W	riteStyleAttr	ibute									
	3	3											
	4	[C#]	public	cons	t H	tmlTextV	VriterAttribute	Style;					
	5	[C++]	public:	con	st H	ItmlText\	WriterAttribute	Style;					
	6	[VB]	Public	Const	Style	As	HtmlTextWr	iterAttribute					
	7	[JScript]	public	var	Style	:	HtmlTextWri	terAttribute;					
	8												
	9	Descripti	on										
	10	Sp	ecifies that the	he HTML <b>st</b>	t <b>yle</b> attribut	e should	be written to the	tag.					
	11	W	riteStyleAttri	bute									
	12												
	13	[C#]	public	const	Html'	TextWrite	erAttribute	Tabindex;					
H	14	[C++]	public:	const	Html	TextWrit	erAttribute	Tabindex;					
1 all	15	[VB]	Public	Const	Tabindex	As	HtmlTextWri	terAttribute					
	16	[JScript]	public	var	Tabindex	:	HtmlTextWrit	erAttribute;					
, <b>.</b>	17												
	18	Description	on										
	19	Spe	ecifies that th	ne HTML <b>ta</b>	bindex attr	ibute sho	uld be written to	the tag.					
	20	Wı	riteStyleAttri	bute									
	22	[C#]	public	const	Htm	nlTextWr	iterAttribute	Target;					
	23	[C++]	public:	const	Htr	nlTextW1	riterAttribute	Target;					
	24	[VB]	Public	Const	Target	As	HtmlTextWri	terAttribute					
	25	[JScript]	public	var	Target	:	HtmlTextWrite	erAttribute;					

lee@hayes plic 509+324+9256

1143

MS1-863US.APP

1												
2	Descripti	on										
3	Sp	ecifies that th	e HTML tar	<b>get</b> attri	bute should	d be written to the	tag.					
4	$\mathbf{W}_{1}$	riteStyleAttril	oute									
5	5											
6	6 [C#] public const HtmlTextWriterAttribute											
7	[C++]	public:	const	:	HtmlText	VriterAttribute	Title;					
8	[VB]	Public	Const	Title	As	HtmlTextWrit	erAttribute					
9	[JScript]	public	var	Title	:	HtmlTextWrite	erAttribute;					
10												
11 12 12 13	Description	on										
'U 12	Spe	ecifies that the	e HTML <b>titl</b>	e attribu	te should b	e written to the tag	g.					
13	Wr	riteStyleAttrib	oute									
14												
15	[C#]	public	const	F	ItmlTextW	riterAttribute	Type;					
16	[C++]	public:	const	]	HtmlTextW	VriterAttribute	Type;					
17	[VB]	Public	Const	Type	As	HtmlTextWrite	erAttribute					
18	[JScript]	public	var	Type	:	HtmlTextWrite	rAttribute;					
19												
20	Descriptio	n e										
21	Spe	ecifies that the	e HTML typ	e attribu	te should b	e written to the tag	g.					
22	Wr	iteStyleAttrib	ute									
23												
24	[C#]	public	const	Ht	mlTextWri	terAttribute	Valign;					
25	[C++]	public:	const	Ht	tmlTextWr	iterAttribute	Valign;					

	1	[VB]	Public	Const	Valign	As	HtmlTextWri	terAttribute
	2	[JScript]	public	var	Valign	:	HtmlTextWrite	erAttribute;
	3							
	4	Description	on					
	5	Sp	ecifies that th	ne HTML <b>va</b>	<b>lign</b> attribu	ite should	d be written to the	tag.
	6		riteStyleAttri					
	7							
	8	[C#]	public	const	Htr	nlTextW	riterAttribute	Value;
	9	[C++]	public:	const	Ht	mlTextW	riterAttribute	Value;
1	10	[VB]	Public	Const	Value	As	HtmlTextWrit	erAttribute
	11	[JScript]	public	var	Value	:	HtmlTextWrite	rAttribute;
	12							
	13	Description	on					
Ħ	14	Spe	ecifies that th	e HTML va	<b>lue</b> attribut	e should	be written to the t	ag.
	15	Wr	iteStyleAttril	oute				
	16							
1 2	17	[C#]	public	const	Htn	nlTextWr	riterAttribute	Width;
	18	[C++]	public:	const	Htr	nlTextW	riterAttribute	Width;
	19	[VB]	Public	Const	Width	As	HtmlTextWrite	erAttribute
:	20	[JScript]	public	var	Width	:	HtmlTextWrite	rAttribute;
:	21							
2	22	Descriptio	n					
1	23	Spe	cifies that the	e HTML wid	lth attribut	e should	be written to the	ag.
2	24	Wri	teStyleAttrib	oute				
2	25							

	1											
	2	∬ [C#]	public	<b>)</b>	const							
	3	[C++]	publi	ic:	const							
	4	[VB]	Public	Cons	st							
	5	[JScript]	publ	ic v	ar							
	6											
	7	Descripi										
	8	Specifies that the HTML wra										
	9	Н	HtmlTextWriterStyle enumera									
1 12/74	10	ToString										
	11											
	12											
	13	Descript	ion									
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	14	$S_1$	pecifies	the	HTM							
1	15	System.	Web.UI.Ht	mlTextW	riter o							
	16	output st	output stream.									
13	17	Н	tmlTextW	riter and	Html3							
	18	values to	register H	ΓML strin	gs to the							
	19	Т	String									
	20											
	21	[C#]	public	const	F							
	22	[C++]	public:	cons	st ]							
	23	[VB]	Public	Const	Backg							
	24	[JScript]	public	var	Back							
	25											
	-											

[C#]	public	const		HtmlTextWr	iterAttribute	Wrap;
[C++]	public:	const		HtmlTextW	riterAttribute	Wrap;
[VB]	Public	Const	Wrap	As	HtmlTextWriterA	attribute
[JScript]	public	var	Wrap	:	HtmlTextWriterA	ttribute;

up attribute should be written to the tag.

ration (System.Web.UI)

ΛL styles available to an or System.Web.UI.Html32TextWriter object

32TextWriter objects use these enumeration ne proper HTML style.

[C#]	public	const	HtmlTextWriter	Style	BackgroundColor;
[C++]	public:	const	HtmlTextWrite	rStyle	BackgroundColor;
[VB]	Public	Const	BackgroundColor	As	HtmlTextWriterStyle
[JScript]	public	var	BackgroundColor	:	HtmlTextWriterStyle;

	1											
	2	Descript	ion									
	3	Sı	pecifies the	HTML ba	a <b>ckgroundcolor</b> style	•						
	4	Т	String									
	5											
	6	[C#]	public	rStyle	BackgroundImage;							
	7	[C++]	public:	erStyle	BackgroundImage;							
	8	[VB]	Public	Const	BackgroundImage	As	HtmlTextWriterStyle					
	9	[JScript]	public	var	BackgroundImage	:	HtmlTextWriterStyle;					
	10											
	11	Description										
	12	Sp	ecifies the	HTML ba	<b>ckgroundimage</b> style	e.						
	13	Тс	String									
	14											
H	15	[C#]	public	const	HtmlTextWri	terStyle	BorderCollapse;					
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	16	[C++]	public:	cons	t HtmlTextWri	terStyle	BorderCollapse;					
	17	[VB]	Public	Const	BorderCollapse	As	HtmlTextWriterStyle					
	18	[JScript]	public	var	BorderCollapse	:	HtmlTextWriterStyle;					
	19											
	20	Description	on									
	21	Sp	ecifies the l	HTML boı	rdercollapse style.							
	22	То	String									
	23											
	24	[C#]	public	cons	t HtmlTextW	riterStyle	BorderColor;					
:	25	[C++]	public:	con	st HtmlTextW	riterStyl	e BorderColor;					

	1 [VB]	Public	Const	BorderColor	As	HtmlTextWriterStyle
	2 [JScript]	public	var	BorderColor	:	HtmlTextWriterStyle;
	3					· ·
	$_{4}\Big\  \ \ Descript$	ion				
	5 S <sub>1</sub>	pecifies the H	HTML bord	<b>ercolor</b> style.		
,		oString		•		
	,    7	-				
;	[C#]	public	const	HtmlTex	tWriterSt	yle BorderStyle;
9	(C++)	public:	cons	t HtmlTex	tWriterS	tyle BorderStyle;
10	[VB]	Public	Const	BorderStyle	As	HtmlTextWriterStyle
) j	[JScript]	public	var	BorderStyle	:	HtmlTextWriterStyle;
	2					
(0 ( <u>]</u> 13	Descripti	ion				
# 14	sp Sp	ecifies the H	TML bord	e <b>rstyle</b> style.		
13) 14] 15	To	String				
1 16	;					
i.⊒ 17	[C#]	public	const	HtmlText	WriterSty	le BorderWidth;
18	[C++]	public:	const	HtmlText	WriterSty	le BorderWidth;
19	[VB]	Public	Const	BorderWidth	As	HtmlTextWriterStyle
20	[JScript]	public	var	BorderWidth	:	HtmlTextWriterStyle;
21						
22	Description	on				
23	Sp	ecifies the H	TML borde	erwidth style.		
24	То	String				
25						

1							
2	[C#]	public	cons	st Ht	mlTextW1	riterStyle	Color;
3	[C++]	public:	con	nst Ht	tmlTextW	riterStyle	Color;
4	[VB]	Public	Const	Color	As	HtmlT	extWriterStyle
5	[JScript]	public	var	Color	:	HtmlTe	xtWriterStyle;
6							
7	Description	on					
8	Sp	ecifies the H7	ΓML color s	style.			
9	То	String					
10							
10 11	[C#]	public	const	HtmlTe	extWriterS	tyle	FontFamily;
12	[C++]	public:	const	HtmlT	extWriterS	Style	FontFamily;
10 11 11 12 13 13	[VB]	Public	Const	FontFamily	As	HtmlTe	extWriterStyle
: 14	[JScript]	public	var	FontFamily	:	HtmlTe	xtWriterStyle;
15							
16	Descriptio	on					
17	Spe	ecifies the HT	ML fontfa	mily style.			
18	To	String					
19							
20	[C#]	public	const	Html	TextWrite:	rStyle	FontSize;
21	[C++]	public:	const	t Html	TextWrite	rStyle	FontSize;
22	[VB]	Public	Const	FontSize	As	HtmlTe	extWriterStyle
23	[JScript]	public	var	FontSize	:	HtmlTex	ktWriterStyle;
24							
25	Descriptio	n					

1	Specifies the HTML fontsize style.											
2	To	ToString										
3												
4	[C#]	public	const	Html	TextWrite	erStyle	FontStyle;					
5	[C++]	public:	cons	st Htm	lTextWrite	erStyle	FontStyle;					
6	[VB]	Public	Const	FontStyle	As	Html	ΓextWriterStyle					
7	[JScript]	public	var	FontStyle	:	HtmlT	extWriterStyle;					
8												
9	Description	on										
10	Specifies the HTML fontstyle style.											
15 15 11	ToString											
12												
10 11 11 12 12 13 13 13 13 13 13 13 13 13 13 13 13 13	[C#]	public	const	HtmlTe	extWriterS	Style	FontWeight;					
14	[C++]	public:	const	HtmlT	extWriter	Style	FontWeight;					
15	[VB]	Public	Const	FontWeight	As	HtmlT	TextWriterStyle					
16	[JScript]	public	var	FontWeight	:	HtmlT	extWriterStyle;					
17												
18	Descriptio	on										
19	Spe	ecifies the H	ΓML fontwo	eight style.								
20	To	String										
21												
22	[C#]	public	cons	t Htn	nlTextWri	terStyle	Height;					
23	[C++]	public:	con	st Htr	nlTextWri	iterStyle	Height;					
24	[VB]	Public	Const	Height	As	HtmlT	extWriterStyle					
25	[JScript]	public	var	Height	:	HtmlTe	extWriterStyle;					

	11										
1											
2	Descripti	ion									
3	Sp	pecifies the I	ITML heig	g <b>ht</b> style.							
4	To	String									
5											
6	[C#]	public	const	HtmlTe	extWriterSty	de Te	xtDecoration;				
7	[C++]	public:	const	HtmlT	extWriterSty	yle Te	xtDecoration;				
8	[VB]	Public	Const	TextDecorat	ion As	HtmlTe	extWriterStyle				
9	[JScript]	public	var	TextDecora	ntion :	HtmlTex	ktWriterStyle;				
10											
11 12 12 13	Descripti	on									
1 <u>1</u> 12	Specifies the HTML <b>textdecoration</b> style.										
道 (章 13	ToString										
<sub>ii</sub> 14											
13 14 15	[C#]	public	co	onst ]	HtmlTextWr	riterStyle	Width;				
16	[C++]	public	: с	onst	HtmlTextW:	riterStyle	Width;				
16	[VB]	Public	Const	Width	As	HtmlTe	xtWriterStyle				
18	[JScript]	public	var	Width	:	HtmlTex	tWriterStyle;				
19	İ										
20	Description	on									
21	Specifies the HTML width style.										
22	HtmlTextWriterTag enumeration (System.Web.UI)										
23	ToString										
24											
25											

### Description

Specifies the HTML tags that can be passed to an System.Web.UI.HtmlTextWriter or System.Web.UI.Html32TextWriter object output stream.

This enumeration allows the output stream to write HTML mark up, along with HTML server controls, in response to a Web request.

**ToString** 

[C#]	public	const	:	HtmlTex	tWriterTag	A;
[C++]	public:	cons	st	HtmlTex	tWriterTag	A;
[VB]	Public	Const	A	As	HtmlTextWr	riterTag
[JScript]	public	var	A	:	HtmlTextWri	terTag;

## Description

Specifies the HTML a element.

**ToString** 

[C#]	public	const	Html	TextWriter1	ag	Acronym;
[C++]	public:	const	Htm	TextWriter	Гад	Acronym;
[VB]	Public	Const	Acronym	As	HtmlTex	tWriterTag
[JScript]	public	var	Acronym	:	HtmlText	WriterTag;

# Description

```
Specifies the HTML acronym element.
    1
              ToString
    2
    3
       [C#]
                     public
                                                 HtmlTextWriterTag
                                    const
                                                                              Address;
       [C++]
                      public:
                                                  HtmlTextWriterTag
                                     const
                                                                              Address;
       [VB]
                   Public
                               Const
                                            Address
                                                          As
                                                                   HtmlTextWriterTag
       [JScript]
                      public
                                             Address
    7
                                   var
                                                                  HtmlTextWriterTag;
    8
       Description
    9
              Specifies the HTML address element.
   10
ToString
   11
   12
       [C#]
                     public
                                                   HtmlTextWriterTag
                                     const
                                                                                Area;
       [C++]
                      public:
                                      const
                                                    HtmlTextWriterTag
                                                                                Area;
       [VB]
                   Public
                                Const
                                             Area
                                                         As
                                                                   HtmlTextWriterTag
       [JScript]
                      public
                                    var
                                              Area
                                                                  HtmlTextWriterTag;
       Description
  18
             Specifies the HTML area element.
  19
             ToString
  20
  21
      [C#]
                     public
  22
                                                     HtmlTextWriterTag
                                      const
                                                                                   В;
      [C++]
                      public:
  23
                                       const
                                                      HtmlTextWriterTag
                                                                                   В;
      [VB]
                   Public
                                 Const
  24
                                              В
                                                        As
                                                                  HtmlTextWriterTag
      [JScript]
                      public
                                               В
                                    var
                                                                 HtmlTextWriterTag;
```

lee@hayes pilc 509+324+9256

1153

MS1-863US.APP

1												
2	Descripti	on										
3	∥ Sp	ecifies the HT	ML <b>b</b> elemen	nt.								
4	То	String										
5												
6	[C#]	public	cons	t	HtmlTextW	riterTag	Base;					
7	[C++]	public:	con	st	HtmlTextW	riterTag	Base;					
8	[VB]	Public	Const	Base	As	HtmlT	extWriterTag					
9	[JScript]	public	var	Base	:	HtmlTe	xtWriterTag;					
10												
11 12 12 13 13 13 15 15 15 15 15 15 15 15 15 15 15 15 15	Description	on										
12	Spe	Specifies the HTML base element.										
13	То	String										
н 14												
15	[C#]	public	const	Ht	mlTextWrite	rTag	Basefont;					
16	[C++]	public:	const	Ht	mlTextWrite	erTag	Basefont;					
17	[VB]	Public	Const	Basefont	As	HtmlTe	extWriterTag					
18	[JScript]	public	var	Basefon	t :	HtmlTe	xtWriterTag;					
19												
20	Descriptio	pn										
21	Spe	Specifies the HTML basefont element.										
22	ToS	String										
23												
24	[C#]	public	const		HtmlTextWi	riterTag	Bdo;					
25	[C++]	public:	cons	st	HtmlTextW	riterTag	Bdo;					

1	[VB]	Public	Const	Bdo	As	HtmlTe	extWriterTag
2	[JScript]	public	var	Bdo	:	HtmlTe	xtWriterTag;
3							_
4	Descriptio	on					
5	Spe	ecifies the HT	ML <b>bdo</b> elei	ment.			
6	To	String					
7							
8	[C#]	public	const	Ht	mlTextWrite	erTag	Bgsound;
9	[C++]	public:	const	Ht	mlTextWrit	erTag	Bgsound;
10	[VB]	Public	Const	Bgsound	As	HtmlTe	extWriterTag
11	[JScript]	public	var	Bgsound	1 :	HtmlTex	xtWriterTag;
11 12 13 13	į						
13	Descriptio	n					
14	Spe	ecifies the HTI	ML bgsound	l element.			
15	ToS	String					
16					,		
i	[C#]	public	cons	t	HtmlTextW	VriterTag	Big;
18	[C++]	public:	con	st	HtmlTextV	VriterTag	Big;
19	[VB]	Public	Const	Big	As	HtmlTe	xtWriterTag
20	[JScript]	public	var	Big	:	HtmlTex	tWriterTag;
21							
22	Description	n					
23	Spe	cifies the HTN	AL big eleme	ent.			
24	ToS	tring					
25							

1							
2	    [C#]	public	const	Htn	nlTextWrite	rTaa	Blockquote;
3	[C++]	public:					-
3		_	const		nlTextWrite	•	Blockquote;
4	[VB]	Public	Const	Blockquo	te As	HtmlT	extWriterTag
5	[JScript]	public	var	Blockqu	ote :	HtmlTe	extWriterTag;
6							
7	Descriptio	on .					
8	Spe	ecifies the HT	ML blockq	<b>uote</b> eleme	ent.		
9	Tos	String					
, 10							
11 12 13 13	[C#]	public	cons	st	HtmlTextV	VriterTag	Body;
12	[C++]	public:	con	nst	HtmlText	_	Body;
13	[VB]	Public	Const	Body	As	_	extWriterTag
14	[JScript]	public	var	Body	:		xtWriterTag;
15		-		J		~ 1 1 1 1	m , , into i rug,
16	Description	n					
16 17	_	cifies the HT	ML hody el	ement			
18		string	Will body of	cinciit.			
	105	umg					
19	50.03						
20	[C#]	public	con	st	HtmlText	WriterTag	Br;
21	[C++]	public:	co	nst	HtmlTex	tWriterTag	Br;
22	[VB]	Public	Const	Br	As	HtmlTe	extWriterTag
23	[JScript]	public	var	Br	:	HtmlTe	xtWriterTag;
24							
25	Description	ı					

1	Sp	pecifies the HI	TML <b>br</b> eleme	ent.								
2	To	String										
3												
4	[C#]	public	const	Htr	nlTextWri	terTag	Button;					
5	[C++]	public:	const	t Hts	mlTextWr	iterTag	Button;					
6	[VB]	Public	Const	Button	As	Html	TextWriterTag					
7	[JScript]	public	var	Button	:	Html	ГехtWriterTag;					
8												
9	Description	on										
10	Sp	Specifies the HTML button element.										
11 12 12 13	То	ToString										
12												
13 13	[C#]	public	const	Htm	lTextWrite	erTag	Caption;					
<sub>ii</sub> 14	[C++]	public:	const	Htm	lTextWrit	erTag	Caption;					
15	[VB]	Public	Const	Caption	As	Html'	TextWriterTag					
15	[JScript]	public	var	Caption	:	HtmlT	extWriterTag;					
17												
18	Descriptio	on										
19		ecifies the HTI	ML <b>caption</b> e	lement.								
20	Tos	String										
21	_											
22	[C#]	public	const	Htm	lTextWrit	erTag	Center;					
23	[C++]	public:	const	Htm	ılTextWrit	terTag	Center;					
24	[VB]	Public	Const	Center	As	Html7	TextWriterTag					
25	[JScript]	public	var	Center	:	HtmlT	extWriterTag;					

1											
2	Description	on									
3	Spe	ecifies the HTI	ML <b>center</b> ele	ment.							
4	То	String									
5											
6	[C#]	public	const		HtmlTextW	/riterTag	Cite;				
7	[C++]	public:	const		HtmlTextV	VriterTag	Cite;				
8	[VB]	Public	Const	Cite	As	HtmlTe	xtWriterTag				
9	[JScript]	public	var	Cite	:	HtmlTex	ktWriterTag;				
10											
10 11	Descriptio	n									
12	Specifies the HTML cite element.										
11 12 13	Tos	ToString									
<sub>ii</sub> 14											
15	[C#]	public	const		HtmlTextWi	riterTag	Code;				
16	[C++]	public:	const		HtmlTextW	riterTag	Code;				
17 4 17	[VB]	Public	Const	Code	As	HtmlTe	xtWriterTag				
18	[JScript]	public	var	Code	:	HtmlTex	tWriterTag;				
19											
20	Description	n									
21	Spec	cifies the HTM	L code eleme	nt.							
22	ToS	tring									
23											
24	[C#]	public	const		HtmlTextW	riterTag	Col;				
25	[C++]	public:	const		HtmlTextW	riterTag	Col;				

1	[VB]	Public	Const	Col	As	HtmlT	extWriterTag
2	[JScript]	public	var	Col	:		xtWriterTag;
3							C.
4	Description	on					
5	    Spo	ecifies the HT	ML <b>col</b> elen	nent.			
6	To	String					
7							
8	[C#]	public	const	Htn	nlTextWrite	rTag	Colgroup;
9	[C++]	public:	const	Htı	mlTextWrite	erTag	Colgroup;
10	[VB]	Public	Const	Colgroup	As	HtmlTe	extWriterTag
11	[JScript]	public	var	Colgroup	· :	HtmlTe	xtWriterTag;
11 11 12 12 13							
13	Descriptio	n					
∷ 14	Spe	cifies the HTI	ML colgrou	<b>p</b> element.			
15	ToS	String					
16							
上 17	[C#]	public	cons	st	HtmlTextW	/riterTag	Dd;
18	[C++]	public:	COI	nst	HtmlTextV	VriterTag	Dd;
19	[VB]	Public	Const	Dd	As	HtmlTe	xtWriterTag
20	[JScript]	public	var	Dd	:	HtmlTex	tWriterTag;
21	_						
22	Description						
23		cifies the HTN	/IL <b>dd</b> eleme	ent.			
24	ToS	tring					
25							

1									
2	[C#]	public	const		HtmlText	WriterTag	Del;		
3	[C++]	public:	cons	t	HtmlText	WriterTag	Del;		
4	[VB]	Public	Const	Del	As	HtmlText	WriterTag		
. 5	[JScript]	public	var	Del	:	HtmlText\	VriterTag;		
6									
7	Description	ı							
8	Specifies the HTML cel element.								
9	ToString								
10									
11	[C#]	public	const		HtmlTextV	VriterTag	Dfn;		
12	[C++]	public:	const		HtmlText	WriterTag	Dfn;		
11 12 13	[VB]	Public	Const	Dfn	As HtmlTex		WriterTag		
: 14	[JScript]	public	var	Dfn	:	HtmlTextV	/riterTag;		
15									
16	Description								
17	Spec	ifies the HTM	IL <b>dfn</b> elemer	nt.					
18	ToSt	ring							
19									
20	[C#]	public	const		HtmlTextV	VriterTag	Dir;		
21	[C++]	public:	const		HtmlTextV	WriterTag	Dir;		
22	[VB]	Public	Const	Dir	As	HtmlTextV	VriterTag		
23	[JScript]	public	var	Dir	:	HtmlTextW	riterTag;		
- 11									
24									

I	Specifies the HTML dir element.								
2	ToString								
3									
4	[C#]	public	const		HtmlText	WriterTag	Div;		
5	[C++]	public:	cons	st	HtmlTex	tWriterTag	Div;		
6	[VB]	Public	Const	Div	As	HtmlText	WriterTag		
7	[JScript]	public	var	Div	:	HtmlTextV	VriterTag;		
8									
9	Description								
10	Sp	ecifies the HTM	AL div eleme	nt.					
11 12 13	То	String							
12									
13	[C#]	public	const		HtmlText	tWriterTag	Dl;		
14	[C++]	public:	cons	st	HtmlTex	tWriterTag	Dl;		
15	[VB]	Public	Const	D1	As	HtmlTextV	VriterTag		
15	[JScript]	public	var	Dl	:	HtmlTextW	<sup>7</sup> riterTag;		
17									
18	Description	on							
19	Spe	ecifies the HTM	IL <b>dl</b> element	••					
20	Tos	String							
21									
22	[C#]	public	const		HtmlText	WriterTag	Dt;		
23	[C++]	public:	cons	t	HtmlText	WriterTag	Dt;		
24	[VB]	Public	Const	Dt	As	HtmlTextW	/riterTag		
25	[JScript]	public	var	Dt	:	HtmlTextW	riterTag;		

1									
2	Description								
3	Specifies the HTML <b>dt</b> element.								
4	ToString								
5									
6	[C#]	public	cons	t	HtmlTextV	VriterTag	Em;		
7	[C++]	public:	con	st	HtmlText	WriterTag	Em;		
8	[VB]	Public	Const	Em	As	HtmlTe	xtWriterTag		
9	[JScript]	public	var	Em	:	HtmlTex	tWriterTag;		
10									
11	Description								
12 13 14	Specifies the HTML em element.								
11 12 13	To	String							
. 14	50.03								
15		public	const		tmlTextWrit	· ·	Embed;		
16	[C++]	public:	const		tmlTextWri	terTag	Embed;		
	[VB]	Public	Const	Embed	As	HtmlTex	tWriterTag		
18	[JScript]	public	var	Embed	:	HtmlTex	tWriterTag;		
19	D								
20	Descriptio								
21		cifies the HTM	IL <b>embed</b> ele	ement.					
22	ToS	String							
23	5000								
24	[C#]	public	const	Htn	nlTextWrite	rTag	Fieldset;		
25	[C++]	public:	const	Htı	mlTextWrite	erTag	Fieldset;		

1	[VB]	Public	Const	Fieldset	As	HtmlTex	xtWriterTag		
2	[JScript]	public	var	Fieldset	:		tWriterTag;		
3									
4	Descriptio	on							
5	Spe	ecifies the HT	ML <b>fieldset</b> el	lement.					
6	Tos	String							
7									
8	[C#]	public	const	Н	tmlTextWri	iterTag	Font;		
9	[C++]	public:	const	·	ItmlTextWr	riterTag	Font;		
_ 10	[VB]	Public	Const	Font	As	HtmlTex	xtWriterTag		
10	[JScript]	public	var	Font	:	HtmlTex	tWriterTag;		
13	Description								
;; 14	Specifies the HTML font element.								
- 15	ToS	String							
16									
17	[C#]	public	const	Ht	mlTextWrit	erTag	Form;		
18	[C++]	public:	const	H	tmlTextWri	terTag	Form;		
19	[VB]	Public	Const	Form	As	HtmlTex	tWriterTag		
20	[JScript]	public	var	Form	:	HtmlText	WriterTag;		
21									
22	Description	n							
23		cifies the HTN	AL <b>form</b> elem	ent.					
24	ToS	tring							
25									

1										
2	[C#]	public	const	Н	tmlTextWri	terTag	Frame;			
3	[C++]	public:	const	t H	ItmlTextWri	iterTag	Frame;			
4	[VB]	Public	Const	Frame	As	HtmlTe	xtWriterTag			
5	[JScript]	public	var	Frame	:	HtmlTex	tWriterTag;			
6										
7	Descriptio	n								
8	Specifies the HTML frame element.									
9	ToString									
10										
11	[C#]	public	const	Htm	nlTextWrite	Tag	Frameset;			
12	[C++]	public:	const	Htr	nlTextWrite	rTag	Frameset;			
	[VB]	Public	Const	Frameset	As	HtmlTe	xtWriterTag			
	[JScript]	public	var	Frameset	:	HtmlTex	tWriterTag;			
14										
16	Descriptio	n								
17	Spe	ecifies the HTI	ML <b>framese</b> t	t element.						
18	ToS	String								
19										
20	[C#]	public	cons	t	HtmlTextW	/riterTag	H1;			
21	[C++]	public:	con	ıst	HtmlTextV	VriterTag	H1;			
22	[VB]	Public	Const	H1	As	HtmlTe	xtWriterTag			
23	[JScript]	public	var	H1	:	HtmlTex	tWriterTag;			
24										
25	Descriptio	n								

lee@hayes plic 509-324-9256

1	Specifies the HTML H1 element.									
2	ToString									
3										
4	[C#]	public	const		HtmlText	tWriterTag	H2;			
5	[C++]	public:	cons	it	HtmlTex	tWriterTag	H2;			
6	[VB]	Public	Const	H2	As	HtmlText	WriterTag			
7	[JScript]	public	var	H2	:	HtmlTextV	VriterTag;			
8										
9	Descriptio	n								
10	Spe	Specifies the HTML H2 element.								
11 11 12 12 13	ToS	String								
12										
13   13	[C#]	public	const		HtmlText	WriterTag	Н3;			
<sub>ii</sub> 14	[C++]	public:	cons	t	HtmlTex	tWriterTag	Н3;			
15	[VB]	Public	Const	Н3	As	HtmlText	WriterTag			
16	[JScript]	public	var	Н3	:	HtmlTextW	/riterTag;			
≟17										
18	Description									
19	_	cifies the HTM	IL <b>H3</b> elemer	nt.						
20	ToS	String								
21										
22	[C#]	public	const		HtmlText	WriterTag	H4;			
23	[C++]	public:	const		HtmlText	WriterTag	H4;			
24	[VB]	Public	Const	H4	As	HtmlTextV	VriterTag			
25	[JScript]	public	var	H4	:	HtmlTextW	riterTag;			

11										
1										
2	Description	ı								
3	Specifies the HTML <b>H4</b> element.									
4	ToString									
5										
6	[C#]	public	const		HtmlTextW	riterTag	H5;			
7	[C++]	public:	const		HtmlTextV	VriterTag	H5;			
8	[VB]	Public	Const	H5	As	HtmlText	extWriterTag			
9	[JScript]	public	var	H5	:	HtmlTextV	WriterTag;			
10										
10 11 11 12 12 13	Description	ı								
L 12	Specifies the HTML <b>H5</b> element.									
<b>1</b> 3	ToString									
14										
15	[C#]	public	const		HtmlTextW	riterTag	Н6;			
16	[C++]	public:	const		HtmlTextV	VriterTag	Н6;			
17	[VB]	Public	Const	Н6	As	HtmlText	WriterTag			
18	[JScript]	public	var	Н6	:	HtmlTextV	WriterTag;			
19										
20	Description	ı								
21	Spec	cifies the HTM	IL <b>H6</b> element	t.						
22	ToS	tring								
23										
24	[C#]	public	const		HtmlTextWri	terTag	Head;			
25	[C++]	public:	const		HtmlTextWr	iterTag	Head;			

1	[VB]	Public	Const	Head	As	HtmlTextW1	riterTag	
2	[JScript]	public	var	Head	:	HtmlTextWri	iterTag;	
3								
. 4	Descriptio	n						
5	Spe	cifies the HTN	ML <b>head</b> elem	ent.				
6	ToS	String						
7								
8	[C#]	public	const		HtmlText	WriterTag	Hr;	
9	[C++]	public:	cons	t	HtmlText	WriterTag	Hr;	
10	[VB]	Public	Const	Hr	As	HtmlTextWr	riterTag	
<u>1</u> 11	[JScript]	public	var	Hr	:	HtmlTextWri	terTag;	
10 11 11 12 13 14 14 15	Specifies the HTML <b>hr</b> element.  ToString							
17	[C#]	public	const		HtmlTextWr	iterTag	Html;	
18	[C++]	public:	const		HtmlTextW	riterTag	Html;	
19	[VB]	Public	Const	Html	As	HtmlTextWr	iterTag	
20	[JScript]	public	var	Html	:	HtmlTextWri	terTag;	
21								
22	Description	n						
23	Spe	cifies the HTM	IL <b>html</b> eleme	ent.				
24	ToS	tring						
25								

1							
2	[C#]	public	cons	st	HtmlTex	tWriterTag	I;
3	[C++]	public:	con	nst	HtmlTe	xtWriterTag	I;
4	[VB]	Public	Const	I	As	HtmlText	WriterTag
5	[JScript]	public	var	I	:	HtmlText	WriterTag;
6							
7	Description	n					
8	Spe	cifies the HTM	ИL <b>i</b> element.				
9	ToS	String					
10							
<u>[</u> 11	[C#]	public	const	Н	tmlTextWri	iterTag	Iframe;
112	[C++]	public:	const	F	ItmlTextWr	riterTag	Iframe;
10	[VB]	Public	Const	Iframe	As	HtmlText	WriterTag
14	[JScript]	public	var	Iframe	:	HtmlTextV	VriterTag;
"=15							
116	Description	n					
17	Spe	cifies the HTM	IL <b>iframe</b> ele	ement.			
18	ToS	tring					
19							
20	[C#]	public	const		HtmlTextW	riterTag	Img;
21	[C++]	public:	const		HtmlTextV	VriterTag	Img;
22	[VB]	Public	Const	Img	As	HtmlText	WriterTag
23	[JScript]	public	var	Img	:	HtmlTextV	VriterTag;
24							
25	Description	ı					

1	Spe	ecifies the HT	ML <b>img</b> elem	nent.					
2	To	String							
3									
4	[C#]	public	const	- -	HtmlTextW	riterTag	Input;		
5	[C++]	public:	cons	t	HtmlTextW	VriterTag	Input;		
6	[VB]	Public	Const	Input	As	HtmlTe	xtWriterTag		
7	[JScript]	public	var	Input	:	HtmlTe	xtWriterTag;		
8									
9	Descriptio	n							
10	Spe	cifies the HTI	ML <b>input</b> ele	ment.					
110	ToString								
12	·								
<b>1</b> 3	[C#]	public	const	t	HtmlText	WriterTag	Ins;		
	[C++]	public:	cons	st	HtmlText	WriterTag	Ins;		
14	[VB]	Public	Const	Ins	As	HtmlTe	xtWriterTag		
16	[JScript]	public	var	Ins	:	HtmlTex	tWriterTag;		
17									
18	Descriptio	n							
19	Spe	cifies the HTN	ML ins eleme	ent.					
20	ToS	String							
21									
22	[C#]	public	const	Ht	mlTextWri	terTag	Isindex;		
23	[C++]	public:	const	H	tmlTextWri	iterTag	Isindex;		
24	[VB]	Public	Const	Isindex	As	HtmlTe	xtWriterTag		
25	[JScript]	public	var	Isindex	:	HtmlTex	tWriterTag;		

```
1
      Description
   2
             Specifies the HTML isindex element.
   3
             ToString
   5
      [C#]
                     public
                                                                                   Kbd;
                                      const
                                                     HtmlTextWriterTag
   6
      [C++]
                      public:
                                                     HtmlTextWriterTag
   7
                                       const
                                                                                   Kbd;
      [VB]
                   Public
                                                                    HtmlTextWriterTag
                                 Const
                                              Kbd
  8
                                                          As
      [JScript]
                      public
                                               Kbd
                                                                    HtmlTextWriterTag;
  9
                                    var
10
111
12
12
      Description
             Specifies the HTML kbd element.
             ToString
·Ç
<sub>ii</sub> 14
15
      [C#]
                    public
                                                    HtmlTextWriterTag
                                     const
                                                                                 Label;
: alla
16
      [C++]
                     public:
                                                    HtmlTextWriterTag
                                      const
                                                                                 Label;
⊫17
      [VB]
                  Public
                                Const
                                             Label
                                                                    HtmlTextWriterTag
                                                          As
      [JScript]
                      public
                                              Label
                                   var
 18
                                                                   HtmlTextWriterTag;
 19
      Description
 20
             Specifies the HTML label element.
 21
            ToString
 22
 23
      [C#]
                    public
                                                  HtmlTextWriterTag
 24
                                   const
                                                                               Legend;
                     public:
                                     const
                                                   HtmlTextWriterTag
                                                                               Legend;
```

1	[VB]	Public	Const	Legend	As	HtmlTex	xtWriterTag
	[JScript]	public	var	Legend			tWriterTag;
2	[Joenpt]	puone	vai	Legenu	•	rumrex	i willer rag,
3							
4	Descriptio	n					
5	Spe	cifies the HTN	ML <b>legend</b> ele	ement.			
6	ToS	String					
7							
8	[C#]	public	const		HtmlText	WriterTag	Li;
9	[C++]	public:	cons	st	HtmlTex	tWriterTag	Li;
10	[VB]	Public	Const	Li	As	HtmlTex	tWriterTag
	[JScript]	public	var	Li	:	HtmlText	tWriterTag;
12							
	Descriptio	n					
# 14	Spe	cifies the HTN	AL <b>li</b> element				
- - 15	ToS	String					
16							
17	[C#]	public	const		HtmlTextW	riterTag	Link;
18	[C++]	public:	const		HtmlTextV	C	Link;
19	[VB]	Public	Const	Link	As	_	tWriterTag
20	[JScript]	public	var	Link			_
.,	[350Hpt]	puone	vai	Lilik	:	Humitex	:WriterTag;
21	Danamintin						
22	Description						
23		cifies the HTN	AL <b>link</b> eleme	ent.			
24	ToS	tring					
25							

1							
2	[C#]	public	const	I	HtmlTextW	riterTag	Мар;
3	[C++]	public:	cons	st .	HtmlTextW	riterTag	Мар;
4	[VB]	Public	Const	Map	As	HtmlTe	extWriterTag
5	[JScript]	public	var	Map	:	HtmlTex	xtWriterTag;
6							
7	Descriptio	n					
8	Spe	cifies the HT	ML <b>map</b> elen	nent.			
9	ToS	String					
10							
11	[C#]	public	const	Htm	lTextWrite	rTag	Marquee;
10	[C++]	public:	const	Htn	nlTextWrite	erTag	Marquee;
13	[VB]	Public	Const	Marquee	As	HtmlTe	xtWriterTag
14	[JScript]	public	var	Marquee	:	HtmlTex	xtWriterTag;
15							
16	Descriptio	n					
17	Spe	cifies the HTI	ML marquee	element.			
18	ToS	String					
19							
20	[C#]	public	const	H	tmlTextWri	terTag	Menu;
21	[C++]	public:	const	Н	tmlTextWr	iterTag	Menu;
22	[VB]	Public	Const	Menu	As	HtmlTe	xtWriterTag
23	[JScript]	public	var	Menu	:	HtmlTex	tWriterTag;
24							
25	Description	n					

1	Specifies the HTML menu element.									
2	ToString									
3										
4	[C#]	public	cons	t H	tmlTextW	riterTag	Meta;			
5	[C++]	public:	con	st H	[tmlTextW	/riterTag	Meta;			
6	[VB]	Public	Const	Meta	As	HtmlTe	extWriterTag			
7	[JScript]	public	var	Meta	:	HtmlTe	xtWriterTag;			
8										
9	Descriptio	on								
10	Spe	ecifies the HT	ML <b>meta</b> ele	ement.						
	ToString									
12										
	[C#]	public	cons	t H	tmlTextW	riterTag	Nobr;			
14	[C++]	public:	con	st H	(tmlTextW	/riterTag	Nobr;			
15	[VB]	Public	Const	Nobr	As	HtmlTe	extWriterTag			
116	[JScript]	public	var	Nobr	:	HtmlTe	xtWriterTag;			
i = 17										
18	Descriptio	on								
19	Spe	ecifies the HT	ML nobr ele	ement.						
20	Tos	String								
21										
22	[C#]	public	const		ΓextWrite:	Ü	Noframes;			
23	[C++]	public:	const		TextWrite	_	Noframes;			
24	[VB]	Public	Const	Noframes	As		extWriterTag			
25	[JScript]	public	var	Noframes	:	HtmlTe	xtWriterTag;			

1									
2	Descriptio	n							
3	Spe	cifies the HT	ML <b>nofram</b>	es element.					
4	ToS	String							
5									
6	[C#]	public	const	Htm	lTextWrite:	rTag	Noscript;		
7	[C++]	public:	const	t Htm	lTextWrite	erTag	Noscript;		
8	[VB]	Public	Const	Noscript	As	HtmlTex	xtWriterTag		
9	[JScript]	public	var	Noscript	:	HtmlTex	tWriterTag;		
10									
# 11	Description	on							
10	Specifies the HTML <b>noscript</b> element.								
13 l	ToString								
<sub>#</sub> 14	1					_	01.1		
15	[C#]	public	cons		mlTextWri		Object;		
16	[C++]	public:	con	st Ht	mlTextWr		Object;		
≟ 17	[VB]	Public	Const	Object	As	HtmlTe	xtWriterTag		
18	[JScript]	public	var	Object	:	HtmlTex	tWriterTag;		
19									
20	Descriptio	on							
21	Sp	ecifies the HT	ML object	element.					
22	То	String							
23									
24	[C#]	public	co	nst	HtmlText	WriterTag	Ol;		
25	[C++]	public:	c	onst	HtmlText	WriterTag	Ol;		

1	[VB]	Public	Const	Ol	As	HtmlText	WriterTag
2	[JScript]	public	var	Ol	:	HtmlText	WriterTag;
3							
4	Description	n					
5	Spe	cifies the HTM	IL ol elemen	ıt.			
6	ToS	string					
7							
8	[C#]	public	const		nlTextWri		Option;
9	[C++]	public:	const	Ht	mlTextWr		Option;
10	[VB]	Public	Const	Option	As		tWriterTag
11 12 11	[JScript]	public	var	Option	:	HtmlText	WriterTag;
10							
13	Description		MI ontion o	lamant			
14		ecifies the HTI	MT option e	iement.			
15	10	String					
11.	[ [CH]	muhlia	cor	net	HtmlTex	ctWriterTag	Ρ;
17	ra	public public:		onst		xtWriterTag	P;
18	[] (XXD) .	Public	Const	Р	As		xtWriterTag
19	[TC aminst]	public	var	P	:		tWriterTag;
20		paone	V 442				
2	n	0n					
2:	C	ecifies the HT	'ML <b>n</b> eleme	nt.			
2	T.	String	TVIL P CICING				
2		Bumg					
2	5						

1							
2	[C#]	public	const	Н	tmlTextWri	terTag	Param;
3	[C++]	public:	const	ŀ	ItmlTextWr	iterTag	Param;
4	[VB]	Public	Const	Param	Param As HtmlTe		WriterTag
5	[JScript]	public	var	Param	:	HtmlTextV	VriterTag;
6							
7	Description	ı					
8	Spec	cifies the HTN	/IL <b>param</b> el	ement.			
9	ToS	tring					
10							
10	[C#]	public	const		HtmlTextV	VriterTag	Pre;
12	[C++]	public:	cons	st	HtmlText	WriterTag	Pre;
1213	[VB]	Public	Const	Pre	As	HtmlText	WriterTag
# 14	[JScript]	public	var	Pre	:	HtmlTextV	VriterTag;
15							
16	Description	n					
17	Spe	cifies the HTN	ML <b>pre</b> elem	ent.			
18	ToS	tring					
19							_
20	[C#]	public	cons			tWriterTag	Q;
21	[C++]	public:	con			tWriterTag	Q;
22	[VB]	Public	Const	Q	As		WriterTag
23	[JScript]	public	var	Q	:	HtmlText	Writer Fag;
24							
25	Descriptio	n					

1	Spe	cifies the HTM	IL q element.				
2	ToS	String					
3							
4	[C#]	public	const		HtmlText	WriterTag	Rt;
5	[C++]	public:	const		HtmlTex	tWriterTag	Rt;
6	[VB]	Public	Const	Rt	As	HtmlText	WriterTag
7	[JScript]	public	var	Rt	:	HtmlTextV	VriterTag;
8							
9	Descriptio	n					
=10	Spe	cifies the HTM	IL <b>rt</b> element.				
10 11 11 11 11 11 11 11 11 11 11	ToS	String					
12							
[]13 []	[C#]	public	const		HtmlTextW	riterTag	Ruby;
: 14	[C++]	public:	const		HtmlTextW	riterTag	Ruby;
#15	[VB]	Public	Const	Ruby	As	HtmlText	WriterTag
16	[JScript]	public	var	Ruby	:	HtmlTextV	VriterTag;
17							
18	Description	n					
19	Spe	cifies the HTM	IL <b>ruby</b> eleme	ent.			
20	ToS	string					
21							
22	[C#]	public	const		HtmlTex	tWriterTag	S;
23	[C++]	public:	const	-	HtmlTex	tWriterTag	S;
24	[VB]	Public	Const	S	As	HtmlText	WriterTag
25	[JScript]	public	var	S	:	HtmlTextV	VriterTag;

1									
2	Description	!							
3	Spec	ifies the HTM	IL <b>s</b> element.						
4	ToSt	tring							
5									
6	[C#]	public	const	H	tmlTextWri	terTag	Samp;		
7	[C++]	public:	const	Н	[tmlTextWr	iterTag	Samp;		
8	[VB]	Public	Const	Samp	As	HtmlTe	xtWriterTag		
9	[JScript]	public	var	Samp	:	HtmlTex	tWriterTag;		
10									
10 111 112 113	Description	n							
12	Spe	cifies the HTN	ML <b>samp</b> elen	nent.					
13	ToString								
:: 14									
15	[C#]	public	const	Н	tmlTextWr	iterTag	Script;		
16	[C++]	public:	const	H	ItmlTextW1	riterTag	Script;		
17	[VB]	Public	Const	Script	As	HtmlTe	extWriterTag		
18	[JScript]	public	var	Script	:	HtmlTe	xtWriterTag;		
19									
20	Description	on							
21	Spe	ecifies the HT	ML <b>script</b> ele	ment.					
22	Tos	String							
23									
24	[C#]	public	const	H	ItmlTextWi	riterTag	Select;		
25	[C++]	public:	const	·	HtmlTextW	riterTag	Select;		

1	[VB]	Public	Const	Select	As	HtmlTex	tWriterTag
2	[JScript]	public	var	Select	:	HtmlText	WriterTag;
3							
4	Descriptio	n					
5	Spe	cifies the HTM	ML <b>select</b> elen	nent.			
6	ToS	String					
7							
8	[C#]	public	const	Н	tmlTextWrit	erTag	Small;
9	[C++]	public:	const	ŀ	łtmlTextWri	terTag	Small;
10	[VB]	Public	Const	Small	As	HtmlTex	tWriterTag
	[JScript]	public	var	Small	:	HtmlText	WriterTag;
11 12 13							
13	Descriptio	n					
14	Spe	cifies the HTM	ML <b>small</b> elen	nent.			
15	ToS	String					
16							
17	[C#]	public	const	Ŧ	HtmlTextWri	terTag	Span;
18	[C++]	public:	const	]	HtmlTextWr	iterTag	Span;
19	[VB]	Public	Const	Span	As	HtmlTex	tWriterTag
20	[JScript]	public	var	Span	:	HtmlText	WriterTag;
21							
22	Descriptio	n					
23	Spe	cifies the HTN	ML <b>span</b> elem	ent.			
24	ToS	String					
25							

1							
2	[C#]	public	const	Ht	mlTextWrit	erTag	Strike;
3	[C++]	public:	const	H	tmlTextWri	terTag	Strike;
4	[VB]	Public	Const	Strike	As	HtmlTex	ktWriterTag
5	[JScript]	public	var	Strike	:	HtmlTex	tWriterTag;
6							
7	Description	n					
8	Spec	cifies the HTN	AL <b>strike</b> eler	ment.			
9	ToS	tring					
10							
111	[C#]	public	const	Ht	mlTextWrit	erTag	Strong;
12	[C++]	public:	const	H	tmlTextWri	terTag	Strong;
13	[VB]	Public	Const	Strong	As	HtmlTe	xtWriterTag
	[JScript]	public	var	Strong	:	HtmlTex	ktWriterTag;
14							
16	Descriptio	n					
17	Spe	cifies the HT	ML <b>strong</b> ele	ement.			
18	Tos	String					
19							a. 1
20	[C#]	public	const		ItmlTextW1	_	Style;
21	[C++]	public:	const		HtmlTextW		Style;
22	[VB]	Public	Const	Style	As		extWriterTag
23	[JScript]	public	var	Style	:	HtmlTe	xtWriterTag;
24							
25	Description	on					

1	Specifies the HTML style element.											
2	ToS	String										
3												
4	[C#]	iterTag	Sub;									
5	[C++]	public:	const		HtmlTextW	riterTag	Sub;					
6	[VB]	Public	Const	Sub	As	HtmlTe	xtWriterTag					
7	[JScript]	public	var	Sub	:	HtmlTex	tWriterTag;					
8												
9	Description											
10	Specifies the HTML sub element.											
	ToS	string										
12												
110 111 12 13	[C#]	public	const		HtmlTextWr	iterTag	Sup;					
	[C++]	public:	const		HtmlTextW	riterTag	Sup;					
14 [[	[VB]	Public	Const	Sup	As	HtmlTe	xtWriterTag					
16	[JScript]	public	var	Sup	:	HtmlTex	tWriterTag;					
17												
18	Description	n										
19	Spe	cifies the HTN	AL sup elemen	nt.								
20	ToS	String										
21												
22	[C#]	public	const	]	HtmlTextWrit	erTag	Table;					
23	[C++]	public:	const		HtmlTextWri	terTag	Table;					
24	[VB]	Public	Const	Table	As	HtmlTe	xtWriterTag					
25	[JScript]	public	var	Table	:	HtmlTex	tWriterTag;					

1										
2	   Descriptio	n								
3	Specifies the HTML <b>table</b> element.									
4	ToString									
5										
6	[C#] public const HtmlTextWriterTag						Tbody;			
7	[C++]	public:	const	Н	tmlTextWr	iterTag	Tbody;			
8	[VB]	Public	Const	Tbody	As	HtmlTe	xtWriterTag			
9	[JScript]	public	var	Tbody	:	HtmlTex	tWriterTag;			
10										
	Descriptio	n								
12	Specifies the HTML <b>tbody</b> element.									
10 11 11 12 13 13	ToString									
8										
14	[C#]	public	const		HtmlText	WriterTag	Td;			
<b>1</b> 6	[C++]	public:	cons	t	HtmlText	WriterTag	Td;			
17	[VB]	Public	Const	Td	As	HtmlTe	xtWriterTag			
18	[JScript]	public	var	Td	:	HtmlTex	tWriterTag;			
19										
20	Descriptio	n								
21	Spe	ecifies the HTM	ML <b>td</b> elemen	t.						
22	ToS	String								
23										
24	[C#]	public	const	Htm	nlTextWrite	erTag	Textarea;			
25	[C++]	public:	const	Htr	nlTextWrit	erTag	Textarea;			

1	[VB]	Public	Const	Textarea	As	HtmlTe	xtWriterTag
2	[JScript]	public	var	Textarea	:	HtmlTex	tWriterTag;
3							
4	Descriptio	n					
5	Spe	ecifies the HT	ML textare	a element.			
6	Tos	String					
7							
8	[C#]	public	cons	t H	ItmlTextWr	riterTag	Tfoot;
9	[C++]	public:	con	st I	HtmlTextW	riterTag	Tfoot;
	[VB]	Public	Const	Tfoot	As	HtmlTe	xtWriterTag
11 11 12 12	[JScript]	public	var	Tfoot	:	HtmlTex	tWriterTag;
12							
13	Descriptio	n					
14	Spe	ecifies the HT	ML <b>tfoot</b> ele	ement.			
15	Tos	String					
16							
17	[C#]	public	con	ıst	HtmlText	WriterTag	Th;
18	[C++]	public:	co	nst	HtmlText	WriterTag	Th;
19	[VB]	Public	Const	Th	As	HtmlTe	xtWriterTag
20	[JScript]	public	var	Th	:	HtmlTex	tWriterTag;
21							
22	Descriptio	n .					
23	Spe	ecifies the HTI	ML <b>th</b> elem	ent.			
24	Tos	String					
25							

. 11							
2	[C#]	public	const	Ht	mlTextWrit	erTag	Thead;
3	[C++]	public:	const	H	tmlTextWri	terTag	Thead;
4	2	Public	Const	Thead	As	HtmlTextV	/riterTag
5	[JScript]	public	var	Thead	:	HtmlTextW	riterTag;
6							
7	Description						
8	Spec	ifies the HTN	ML <b>thead</b> ele	ment.			
9	ToSt	ring					
10							
10 111 112 13	[C#]	public	const		HtmlTextW		Title;
112 12	[C++]	public:	cons	t	HtmlTextW		Title;
13 13	[VB]	Public	Const	Title	As	HtmlText	
	[JScript]	public	var	Title	:	HtmlText\	VriterTag;
14 15 16							
	Description						
4 17	Spe	cifies the HT	ML title elen	nent.			
18	ToS	String					
19	,						75
20	∫ [C#]	public	con	st		tWriterTag	Tr;
2	[C++]	public	: co	nst	HtmlTex	tWriterTag	Tr;
2	[VB]	Public	Const	Tr	As		tWriterTag
2	[JScript]	public	var	Tr	:	HtmlText	WriterTag;
2	4						
2	5 Description	on					

1	Specifies the HTML tr element.										
2	ToS	tring									
3											
4	[C#]	public	const		HtmlTextWriterTag Tt;						
5	[C++]	public:	const	t	HtmlTex	tWriterTag	Tt;				
6	[VB]	Public	Const	Tt	As	HtmlTextW	riterTag				
7	[JScript]	public	var	Tt	:	HtmlTextWi	riterTag;				
8											
9	Description	$\eta$									
10	Specifies the HTML tt element.										
10	ToString										
12											
1	[C#]	public	const		HtmlTex	tWriterTag	U;				
14	[C++]	public:	cons	t	HtmlTex	ctWriterTag	U;				
15	[VB]	Public	Const	U	As	HtmlTextW	riterTag				
16	[JScript]	public	var	U	:	HtmlTextWi	riterTag;				
17											
18	Description	r									
19	•	cifies the HTM	L <b>u</b> element.								
20	ToS	tring									
21											
22	[C#]	public	const			WriterTag	Ul;				
23	[C++]	public:	const			tWriterTag	Ul;				
24	[VB]	Public	Const	Ul	As	HtmlTextW					
25	[JScript]	public	var	U1	:	HtmlTextWi	riterTag;				

Descriptio	Description								
Specifies the HTML ul element.									
ToString									
[C#]	public	const	Htn	nlTextWrite	rTag	Unknown;			
[C++]	public:	const	Htr	nlTextWrite	erTag	Unknown;			
[VB]	Public	Const	Unknown	As	HtmlTe	extWriterTag			
[JScript]	public	var	Unknow	n :	HtmlTe	xtWriterTag;			
Description									
The	System.Stri	ng passed as	an HTML	tag is not re	ecognized.				
Description  The System.String passed as an HTML tag is not recognized.  ToString									
[C#]	public	con	st	HtmlTextV	VriterTag	Var;			
[C++]	public:	CO	nst	HtmlText	WriterTag	Var;			
[VB]	Public	Const	Var	As	HtmlTe	extWriterTag			
[JScript]	public	var	Var	:	HtmlTe	xtWriterTag;			
Descriptio	n								
Spe	cifies the HT	ML <b>var</b> eler	nent.						
ToString									
[C#]	public	cons	st	HtmlTextW	VriterTag	Wbr;			
	Spe ToS  [C#] [C++] [VB] [JScript]  Descriptio The ToS  [C#] [C++] [VB] [JScript]  Descriptio Spe	Specifies the HT ToString  [C#] public [C++] public: [VB] Public [JScript] public  Description  The System.Stri ToString  [C#] public [C++] public: [VB] Public [JScript] public  Description  Specifies the HT	Specifies the HTML ul element ToString  [C#] public const [C++] public: const [VB] Public Const [JScript] public var  Description The System.String passed as ToString  [C#] public const [C++] public: const [VB] Public Const [JScript] public var  Description Specifies the HTML var element	Specifies the HTML ul element.  ToString  [C#] public const Htm [C++] public: const Htm [VB] Public Const Unknown [JScript] public var Unknown  Description  The System.String passed as an HTML ToString  [C#] public const [C++] public: const [VB] Public Const Var [JScript] public Var Var  Description  Specifies the HTML var element.	Specifies the HTML ul element.  ToString  [C#] public const HtmlTextWrite [C++] public: const HtmlTextWrite [VB] Public Const Unknown As [JScript] public var Unknown :  Description  The System.String passed as an HTML tag is not reserved.  ToString  [C#] public const HtmlTextV [C++] public: const HtmlTextV [VB] Public Const Var As [JScript] public var Var :  Description  Specifies the HTML var element.	Specifies the HTML ul element.  ToString  [C#] public const HtmlTextWriterTag  [C++] public: const HtmlTextWriterTag  [VB] Public Const Unknown As HtmlText  [JScript] public var Unknown : HtmlText  Description  The System.String passed as an HTML tag is not recognized.  ToString  [C#] public const HtmlTextWriterTag  [C++] public: const HtmlTextWriterTag  [VB] Public Const Var As HtmlText  [JScript] public var Var : HtmlText  Description  Specifies the HTML var element.			

1	[VB]	Public	Const	Wbr	As	HtmlTextV	VriterTag				
2	[JScript]	public	var	Wbr	:	HtmlTextW	riterTag;				
3											
4	Description										
5	Specifies the HTML wbr element.										
6	ToS	String									
7											
8	[C#]	public	const		HtmlTextV	VriterTag	Xml;				
9	[C++]	public:	cons	t	HtmlText	WriterTag	Xml;				
la l	[VB]	Public	Const	Xml	As	HtmlTextV	VriterTag				
	[JScript]	public	var	Xml	:	HtmlTextW	riterTag;				
111 12 113	Description										
11	Spe	cifies the HTN	ML <b>xml</b> elem	ent.							
14	IAt	tributeAccesso	or interface (S	System.Wo	eb.UI)						
16	ToS	String									
17											
18											
19	Descriptio	n									
20	Def	fines method	s used by	ASP.NI	ET server	controls to	provide				
21	programm	atic access to	any attribu	te declare	ed in the o	opening tag of	a server				
22	control.										
23	If	you author	a custom	server	control th	at inherits f	rom the				
24	-	eb.UI.WebCo					,				
25	System.W	eb.UI.HtmlC	ontrols.Htm	<b>i</b> Control		,	or				

1	System.Web.UI.WebControls.ListItem class, the .NET Framework								
2	automatically provides programmatic access to attributes since each of these								
3	classes implement the IAttributeAccessor interface.								
4	GetAttribute								
5									
6	[C#] string GetAttribute(string key);								
7	[C++] String* GetAttribute(String* key);								
8	[VB] Function GetAttribute(ByVal key As String) As String								
9	[JScript] function GetAttribute(key : String) : String;								
111 112 113	Description  When implemented by a class, retrieves the specified attribute property								
13	from the server control.								
14	Return Value: The value of the specified attribute. A System.String object that								
15	represents the name of the server control attribute.								
16	SetAttribute								
17									
18	[C#] void SetAttribute(string key, string value);								
19	[C++] void SetAttribute(String* key, String* value);								
20	[VB] Sub SetAttribute(ByVal key As String, ByVal value As String)								
21	[JScript] function SetAttribute(key : String, value : String);								
22									
23	Description								
24									

lee@hayes pttc 509+324-9256

	When implemented by a class, designates an attribute and its value to										
1											
2	assign to the ASP.NET server control. The name of the attribute to be set. The										
3	value assigned to the attribute.										
4	IDataBindingsAccessor interface (System.Web.UI)										
5	SetAttribute										
6											
7											
8	Description										
9	Allows access to the collection of databinding expressions on a control at										
10	design time.										
	DataBindings										
12 12	SetAttribute										
12											
-	[C#] DataBindingCollection DataBindings {get;}										
15											
i 15	[C++] DataBindingCollection* get_DataBindings();										
16	[VB] ReadOnly Property DataBindings As DataBindingCollection										
17	[JScript] abstract function get DataBindings() : DataBindingCollection;										
18											
19	Description										
20	Indicates a collection of all data bindings on the control. This property is										
21	read-only.										
22	HasDataBindings										
23	SetAttribute										
24											
25	[C#] bool HasDataBindings {get;}										
1	ı										

19

20

21

22

23

24

25

[C++] bool get\_HasDataBindings();
[VB] ReadOnly Property HasDataBindings As Boolean

[JScript] abstract function get HasDataBindings() : Boolean;

Description

Returns whether the control contains any data binding logic. This method is only accessed by RAD designers.

Return Value: true if the control contains data binding logic; otherwise, false.

ImageClickEventArgs class (System.Web.UI)

SetAttribute

Description

Provides data for any events that occur when a user clicks on a image-based ASP.NET server control, such as the System.Web.UI.HtmlControls.HtmlInputImage or System.Web.UI.WebControls.Image server controls. This class cannot be inherited.

This class passes the location where a user clicked on an HtmlInputImage server control or an Image Web server control. Clicking an HtmlInputImage server control causes a ServerClick event to fire, while clicking an Image server control causes a Click event to occur. You can then use event handlers to programmatically respond to the event based on the value of these coordinates.

SetAttribute

1											
2	[C#]		public			t		Х;			
3	[C++]		public:		ir	ıt		Х;			
4	[VB]	Publi	ic	X		As		Integer			
5	[JScript]	publ	ic	var	X		:	int;			
6											
7	Description										
8	An integer that represents the x-coordinate where a user clicked on an										
9	image-based ASP.NET server control.										
10	SetAttribute										
11											
12	[C#]		public		in	t		Y;			
13	[C++]		public:		ir	nt		Y;			
14	[VB]	Publ	ic	Y		As		Integer			
15	[JScript]	publ	ic	var	Y		:	int;			
16											
17	Description										
18	An int	teger that r	represents th	e y-coordinate	e whe	re a us	er clicked	d on an			
19	image-based.	ASP.NET s	erver control	i.							
20	Image	ClickEvent	Args								
21	Examp	ole Syntax:									
22	SetAttribute										
23											
24	[C#] p	oublic	ImageClic	EventArgs(in	t	х,	int	y);			
25	[C++]	public:	ImageCli	ckEventArgs(i	nt	х,	int	y);			

[VB] Public Sub New(ByVal x As Integer, ByVal y As Integer)

[JScript] public function ImageClickEventArgs(x : int, y : int);

### Description

Initializes a new instance of the **System.Web.UI.ImageClickEventArgs** class. The x-coordinate where the user clicked on an image-based ASP.NET server control. The y-coordinate where the user clicked on an image-based ASP.NET server control.

ImageClickEventHandler delegate (System.Web.UI)
ToString

### Description

Represents the method that will handle any events that are raised when a user clicks on an image-based ASP.NET server control. The server control that is the source of the event. An **System.Web.UI.ImageClickEventArgs** object that contains event data.

This delegate defines the required signature for all click events raised when a user clicks on an image-based ASP.NET server control. You must use the signature defined by this delegate for any OnServerClick event handlers that you create for an System.Web.UI.HtmlControls.HtmlInputImage server control or any OnClick event handlers that you create for an System.Web.UI.WebControls.ImageButton Web server control.

INamingContainer interface (System.Web.UI)
ToString

4

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Identifies a container control that creates a new ID namespace within a **System.Web.UI.Page** object's control hierarchy. This is a marker interface only.

Any control that implements this interface creates a new namespace in which all child control ID attributes are guaranteed to be unique within an entire application. The marker provided by this interface allows unique naming of the dynamically generated server control instances within the Web server controls that the support data binding. These controls include System.Web.UI.WebControls.Repeater the control, System.Web.UI.WebControls.DataGrid control. the or System.Web.UI.WebControls.RadioButtonList control.

IParserAccessor interface (System.Web.UI)
ToString

### Description

Defines the method that ASP.NET server controls must implement to recognize when elements, either HTML or XML, are parsed.

When elements are parsed, they are recognized as children of the server control that implements this interface. As such, they are added to the control's **System.Web.UI.Control.Controls** property. These elements, generally speaking, are added as **System.Web.UI.LiteralControl** objects to the **System.Web.UI.ControlCollection** object associated with **Controls** property.

pe@hayes\_piic 509-334-9256 1193 MSI-863US.APP

### AddParsedSubObject

[C#]	vo	oid	AddParsedSubOl	oject(obje	ct	obj);
[C++]	V	oid	AddParsedSubOb	ject(Obje	ct*	obj);
[VB]	Sub	AddPar	rsedSubObject(ByVal	obj	As	Object)
[JScript]	fun	ction	AddParsedSubObjec	t(obj	:	Object);

### Description

When implemented by an ASP.NET server control, notifies the server control that an element, either XML or HTML, was parsed, and adds the element to the server control's **System.Web.UI.ControlCollection** object. The **System.Object** instance to add to the parent control.

IPostBackDataHandler interface (System.Web.UI)
AddParsedSubObject

### Description

Defines methods that ASP.NET server controls must implement to automatically load post back data.

If you want a server control you design to examine form data that is posted back to the server by the client, you must implement the **IPostBackDataHandler** interface. The contract that this interface defines allows a server control to determine whether its state should be altered as a result of the post back, and to raise the appropriate events.

LoadPostData

[C#]	bool	LoadPostData	a(string	postData	Key,	Nam	eValue(	Collection
postCol	lection)	;						
[C++]	bool	LoadPostData	(String*	postData	Key,	Name	ValueCo	ollection*
postCol	lection)	•						
[VB]	Function	n LoadPostD	ata(ByVal	postDa	ıtaKey	As	String,	ByVal
postCol	lection	As	NameVa	lueCollec	tion)	A	As	Boolean
[JScript	[] func	tion LoadPost	tData(postl	DataKey	: Sta	ring,	postColl	ection :
NameValueCollection)				:				Boolean;

## Description

When implemented by a class, processes post back data for an ASP.NET server control.

Return Value: true if the server control's state changes as a result of the post back; otherwise false.

The Web Forms page framework tracks all of the server control's that return true to this method call, then invokes the System.Web.UI.IPostBackDataHandler.RaisePostDataChangedEvent on those controls. The key identifier for the control. The collection of all incoming name values.

## RaisePostDataChangedEvent

RaisePostDataChangedEvent();	void	[C#]
RaisePostDataChangedEvent();	void	[C++]
RaisePostDataChangedEvent()	Sub	[VB]

ee⊗hayes pir. 509+324+9256 1195 MS1-863US.APP

19

20

21

22

23

24

[JScript] function RaisePostDataChangedEvent();

Description

2

3

4

5

6

When implemented by a class, signals the server control object to notify the ASP.NET application that the state of the control has changed.

Change events for the server control that implements this interface, if any, are raised from this method.

IPostBackEventHandler interface (System.Web.UI)

RaisePostDataChangedEvent

## Description

Defines the method ASP.NET server controls must implement to handle post back events.

To create a server control that captures form submit information from the browser, you must implement this interface. For more information on how to use this interface, see .

### RaisePostBackEvent

[C#]	void	RaisePostBackEve	nt(string	eventAr	gument);
[C++]	void	RaisePostBackEver	nt(String*	eventAr	gument);
[VB]	Sub RaisePo	stBackEvent(ByVal	eventArgument	As	String)
[JScript]	function	RaisePostBackEvent	t(eventArgument	:	String);
Descripti	ion				

When implemented by a class, enables a server control to process an event raised when a form is posted to the server.

This method provides the functionality for many events implemented by HTML and Web server controls. A **System.String** that represents an optional event argument to be passed to the event handler.

IStateManager interface (System.Web.UI)

RaisePostBackEvent

### Description

Defines the properties and methods any class must implement to support view state management for a server control.

A server control's view state comprises the cumulative values of the control's properties. This interface includes methods that save and load a server control's view state values, as well as a method that instructs the control to track any changes to its view state.

**IsTrackingViewState** 

RaisePostBackEvent

[C#]	bo	ol		IsTrackingViewState		{get;}
[C++]		bool		get_IsTrac	kingV	iewState();
[VB]	ReadOnly	Property		IsTrackingViewState	As	Boolean
[JScript]	abstract	function	get	IsTrackingViewState()	:	Boolean;

Description

When implemented by a class, gets a value indicating whether a server control is tracking its view state changes.

### LoadViewState

[C#]	void	LoadView	State(object		state);
[C++]	void	LoadViewS	State(Object	*	state);
[VB]	Sub Load\	/iewState(ByVal	state	As	Object)
[JScript]	function	LoadViewState	e(state	:	Object);

## Description

When implemented by a class, loads the server control's previously saved view state to the control. An **System.Object** that contains the saved view state values for the control.

#### SaveViewState

[C#]		object	Save	ViewState();
[C++]		Object*	Save	ViewState();
[VB]	Function	SaveViewState()	As	Object
[JScript]	function	SaveViewState()	:	Object;

# Description

When implemented by a class, saves the changes to a server control's view state to an System.Object .

Return Value: The Object that contains the view state changes.

TrackViewState

[C#]	void	TrackViewState();
[C++]	void	TrackViewState();
[VB]	Sub	TrackViewState()
[IScript]	function	TrackViewState();

When implemented by a class, instructs the server control to track changes to its view state.

Once this method has been called on a server control, the System.Web.UI.IStateManager.IsTrackingViewState property will return true

ITagNameToTypeMapper interface (System.Web.UI)
TrackViewState

### Description

Maps a sequence of text characters to a .NET Framework type when an .aspx file is processed on the server.

Classes that implement this interface strongly type any string in an .aspx file to the appropriate class. HTML and literal text strings are mapped to the type System.Web.UI.LiteralControl , HTML anchor tags with a runat="server" attribute/value pair are mapped to the type System.Web.UI.HtmlControls.HtmlAnchor , while tags are mapped to the type System.Web.UI.WebControls.Button .

## GetControlType

tagName, **IDictionary** attribs); GetControlType(string [C#] Type IDictionary\* tagName, attribs); GetControlType(String\* [C++]Type\* [VB] Function GetControlType(ByVal tagName As String, ByVal attribs As IDictionary) As Type

[JScript] function GetControlType(tagName : String, attribs : IDictionary) : Type;

## Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Retrieves the .NET Framework type that processes the control declared in the .aspx file.

Return Value: The .NET Framework type that is assigned to the control. The element name of the control sent from the .aspx file. A collection of the attributes on the control in the .aspx file.

ITemplate interface (System.Web.UI)
GetControlType

# Description

Defines the method to implement for populating an ASP.NET server control with child controls when using a the control with inline templates when declared in an .aspx file.

InstantiateIn

[C#] void InstantiateIn(Control container);

container);		ateIn(Control*	Instantia	void		[C++]
Control)	As	container	tiateIn(ByVal	ub Instanti	Sub	[VB]
Control);	:	In(container	InstantiateI	function		[JScript]

When implemented by a class, defines the **System.Web.UI.Control** object that child controls and templates belong to. These child controls are in turn defined within an inline template.

When developing templated server controls you do not need to implement this method, the .NET Framework provides the implementation for you. The **Control** object to contain the instantiated controls from the inline template.

IValidator interface (System.Web.UI)

InstantiateIn

#### Description

Defines the properties and methods that objects that participate in validation must implement.

Classes that implement this interface represent a posssible user input error. When the **System.Web.UI.IValidator.Validate** method is called, the class updates its **System.Web.UI.IValidator.IsValid** property to signify whether the error occurred. The **System.Web.UI.IValidator.ErrorMessage** property contains a text description of the error condition that you can display when the error occurs.

ErrorMessage

InstantiateIn

[C#]	string		ErrorMessage	{get;	set;}
[C++]	String*	get	_ErrorMessage();void	set_ErrorMessage	e(String*);
[VB]	Prope	rty	ErrorMessage	As	String
[JScript] abstract function get ErrorMessage(): String;public abstract function set					
ErrorMessage(String);					

When implemented by a class, gets or sets the error message text generated when the condition being validated fails.

Web Forms page developers access this property declaratively in the opening tag of a validation server control. For more information, see .

### IsValid

### InstantiateIn

[C#]	bool	IsValid	{get;	set;}
[C++]	bool	get_IsValid();void	se	et_IsValid(bool);
[VB]	Property	IsValid	As	Boolean
[JScript] abst	tract function g	et IsValid() : Boolean;p	ublic abstr	act function set
IsValid(Boole	ean);			

## Description

When implemented by a class, gets or sets a value indicating whether the user-entered content in the specified control passes validation.

### Validate

ť,	9	ilinii.
ij		
į,	F	
i,	12	,
THE SHAPE SHAPE	10000	į
ť,	Į,	
į,	;#1	Sans.
ŧ,	11.15	1
;;		
ľ,		
1,	3	1
ļ		
ľ,	# #	i init
ľ, ľ,	::	į.
1		

[C#]	void	Validate();
[C++]	void	Validate();
[VB]	Sub	Validate()
[JScript]	function	Validate();
Description		
When implemen	nted by a class, evaluates the condi	ition it checks and upates
the System.Web.UI.IV	Validator.IsValid property.	
LiteralControl o	class (System.Web.UI)	
Validate		
Description		
Represents HT	ML elements, text, and any other	r strings in an ASP.NET
page that do not requir	re processing on the server.	
ASP.NET com	piles all HTML elements and re	eadable text that do not
require server-side pro	ocessing into instances of this class	s. For example, an HTML

element that does not contain a runat="server" attribute/value pair in its opening tag is compiled into a LiteralControl object.

LiteralControl

Example Syntax:

Validate

LiteralControl(); public [C#]

MS1-863US.APP lee@hayes plic 509-324-9256

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Controls

EnableViewState

LiteralControl(); public: [C++]New() Sub Public [VB] [JScript] public function LiteralControl(); Initializes a new instance of the class. System.Web.UI.LiteralControl Description Initializes a new instance of the System. Web. UI. Literal Control class that contains a literal string to be rendered on the requested ASP.NET page. LiteralControl Example Syntax: Validate LiteralControl(string text); public [C#] LiteralControl(String\* text); public: [C++]String) New(ByVal text As Public Sub [VB] Literal Control (textString); function [JScript] public Description Initializes a new instance of the System. Web. UI. Literal Control class with the specified text. The text to be rendered on the requested Web page. ChildControlsCreated ClientID Context

15

16

17

18

19

20

21

22

23

24

**Events** HasChildViewState 2 ID 3 IsTrackingViewState NamingContainer 5 Page 6 Parent Site **TemplateSourceDirectory** 9 Text 10 Validate 11 12 13

# Description

Gets or sets the text content of the System. Web. UI. Literal Control object.

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Create Control Collection

[C#] protected override ControlCollection CreateControlCollection();
[C++] protected: ControlCollection\* CreateControlCollection();
[VB] Overrides Protected Function CreateControlCollection() As
ControlCollection

[JScript]	protected	override	function	CreateControlCollection()
ControlCol	lection;			

Creates an System.Web.UI.EmptyControlCollection object for the current instance of the System.Web.UI.LiteralControl .

Return Value: The EmptyControlCollection object for the current control.

By default, LiteralControl objects contain only text and no child server controls. If you want to change this behavior, you must override this method.

Render

[C#] protected override void Render(HtmlTextWriter output);
[C++] protected: void Render(HtmlTextWriter\* output);
[VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)
[JScript] protected override function Render(output : HtmlTextWriter);

#### Description

Writes the content of the **System.Web.UI.LiteralControl** object to the ASP.NET page.

When you create a custom server control and want to render specific HTML or text to a client, you can improve the performance of the control by passing the value of the System.Web.UI.LiteralControl.Text property to the System.Web.UI.Control.Render(System.Web.UI.HtmlTextWriter) method rather than calling the LiteralControl.Render method. An

System.Web.UI.HtmlTextWriter object that renders the content of the LiteralControl object to the requesting client.

LosFormatter class (System.Web.UI)

TrackViewState

Description

Serializes the view state for a Web Forms page. The limited object serialization (LOS) formatter is designed for highly compact ASCII format serialization. This class supports serializing any object graph, but is optimized for those containing strings, arrays, and hashtables. It offers second order optimization for many of the .NET primitive types.

This is a private format, and only needs to remain consistent for the lifetime of a Web request. You are not allowed to persist objects serialized with this formatter for any significant length of time.

LosFormatter

Example Syntax:

TrackViewState

[C#] public LosFormatter();
[C++] public: LosFormatter();
[VB] Public Sub New()

[JScript] public function LosFormatter();

Deserialize

lee@hayes pik 509-324-9256 1207 MS1-863US.APP

[C#] public object Deserialize(Stream stream);
[C++] public: Object\* Deserialize(Stream\* stream);
[VB] Public Function Deserialize(ByVal stream As Stream) As Object
[JScript] public function Deserialize(stream : Stream) : Object; Deserializes an Object.

### Description

1

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Descrializes a LOS-formatted object from a **System.IO.Stream** object.

Return Value: Returns the descrialized object. The source of the object to be descrialized.

#### Deserialize

Deserialize(string input); public object [C#] Deserialize(String\* input); public: Object\* [C++][VB] Public Function Deserialize(ByVal input As String) Object Object; function Deserialize(input String) [JScript] public

#### Description

Deserializes a LOS formatted object from a string.

Return Value: Returns the deserialized object. The source of the object to be deserialized.

#### Deserialize

[C#] public object Deserialize(TextReader input);

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

[C++] public: Object\* Deserialize(TextReader\* input);
[VB] Public Function Deserialize(ByVal input As TextReader) As Object
[JScript] public function Deserialize(input : TextReader) : Object;

Description

Deserializes a LOS-formatted object from a System.IO.TextReader object.

Return Value: Returns the descrialized object. The source of the object to be descrialized.

Serialize

object value); Serialize(Stream void stream, public [C#] Object\* value); Serialize(Stream\* stream, public: void [C++][VB] Public Sub Serialize(ByVal stream As Stream, ByVal value As Object) [JScript] public function Serialize(stream : Stream, value : Object); Serializes the value. view state **Forms** Web

### Description

Serializes the Web Forms view state value into a **System.IO.Stream** object. The object to serialize into. The view state information to pass.

Serialize

[C#] public void Serialize(TextWriter output, object value); [C++] public: void Serialize(TextWriter\* output, Object\* value); [VB] Public Sub Serialize(ByVal output As TextWriter, ByVal value As Object)

lee@hayes pilc 509-324-9256 1209

[JScript] public function Serialize(output : TextWriter, value : Object); 2 Description 3 Serializes the view state value into a System.IO.TextWriter object. The 4 object to serialize into. The view state information to pass. 5 ObjectConverter class (System.Web.UI) 6 **ToString** 8 9 Description 10 ObjectConverter 11 Example Syntax: 12 **ToString** 13 14 ObjectConverter(); public [C#] 15 ObjectConverter(); public: [C++]16 New() Sub Public [VB] 17 [JScript] public function ObjectConverter(); 18 ConvertValue 19 20 [C#] public static object ConvertValue(object value, Type toType, string 21 formatString); 22 [C++] public: static Object\* ConvertValue(Object\* value, Type\* toType, String\* 23 formatString); 24 [VB] Public Shared Function ConvertValue(ByVal value As Object, ByVal

```
toType
                             ByVal
                                       formatString
                                                                              Object
                    Type,
              As
                                                       As
                                                             String)
                                                                       As
1
    [JScript] public static function ConvertValue(value : Object, toType : Type,
2
    formatString
                                           String)
                                                                             Object;
3
4
    Description
5
          ObjectTagBuilder class (System.Web.UI)
6
          ToString
7
8
9
    Description
10
          ObjectTagBuilder
11
          Example Syntax:
12
           ToString
13
14
                                 public
                                                                ObjectTagBuilder();
    [C#]
15
    [C++]
                                                                ObjectTagBuilder();
                                 public:
16
    [VB]
                            Public
                                                      Sub
                                                                              New()
17
    [JScript] public function ObjectTagBuilder();
18
          ControlType
19
          FChildrenAsProperties
20
           FIsNonParserAccessor
21
          HasAspCode
22
          ID
23
           InDesigner
24
          NamingContainerType
25
```

1	Parser
2	TagName
3	AppendLiteralString
4	
5	[C#] public override void AppendLiteralString(string s);
6	[C++] public: void AppendLiteralString(String* s);
7	[VB] Overrides Public Sub AppendLiteralString(ByVal s As String)
8	[JScript] public override function AppendLiteralString(s : String);
9	
10	Description
11	AppendSubBuilder
12	
13	[C#] public override void AppendSubBuilder(ControlBuilder subBuilder);
14	[C++] public: void AppendSubBuilder(ControlBuilder* subBuilder);
15	[VB] Overrides Public Sub AppendSubBuilder(ByVal subBuilder As
16	ControlBuilder)
17	[JScript] public override function AppendSubBuilder(subBuilder :
18	ControlBuilder);
19	
20	Description
21	Init
22	
23	[C#] public override void Init(TemplateParser parser, ControlBuilder
24	parentBuilder, Type type, string tagName, string id, IDictionary attribs);
25	[C++] public: void Init(TemplateParser* parser, ControlBuilder* parentBuilder

Type\* String\* id, IDictionary\* attribs); String\* tagName, type, [VB] Overrides Public Sub Init(ByVal parser As TemplateParser, ByVal parentBuilder As ControlBuilder, ByVal type As Type, ByVal tagName As String, IDictionary) ByVal attribs As id String, ByVal As [JScript] public override function Init(parser : TemplateParser, parentBuilder : ControlBuilder, type: Type, tagName: String, id: String, attribs: IDictionary);

### Description

2

3

4

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

OutputCacheLocation enumeration (System.Web.UI)

**ToString** 

### Description

Specifies the valid values for the location of the output cache.

The values specified by this enumeration are used when you include an @ OutputCache directive in a .aspx file. These values set the cacheability of page output. For more information, see .

### **ToString**

[C#]	public	const		OutputCacheLocation		Any;
[C++]	public:	const		OutputCacheLocation		Any;
[VB]	Public	Const	Any	As	OutputCache	Location
[JScript]	public	var	Any	•	OutputCache?	Location;

# Description

lee@hayes pik 509-324-9256 1213 MS1-863US.APP

The output cache can be located on the browser client where the request originated, on a proxy server (or any other server) participating in the request, or on the server where the request was processed.

## **ToString**

[C#]	public	const	O	utputCacheI	Location	Client;
[C++]	public:	const	C	OutputCache	Location	Client;
[VB]	Public	Const	Client	As	OutputCachel	Location
[JScript]	public	var	Client	:	OutputCacheL	ocation;

### Description

The output cache is located on the browser client where the request originated.

## **ToString**

[C#]	public	const	OutputCa	cheLocati	on	Downstream;
[C++]	public:	const	OutputCa	acheLocat	ion	Downstream;
[VB]	Public	Const	Downstream	As	Output	CacheLocation
[JScript]	public	var	Downstream	:	OutputC	CacheLocation;

# Description

The output cache can be stored in any HTTP 1.1 cache-capable devices other than the origin server. This include proxy servers and the client that made the request.

# **ToString**

[C#]	public	const	OutputCacheLocation			None;
[C++]	public:	const	OutputCacheLocation		None;	
[VB]	Public	Const	None	As	OutputCacheI	Location
[JScript]	public	var	None	:	OutputCacheL	ocation;

The output cache is disabled for the requested page.

**ToString** 

[C#]	public	const	Ou	tputCacheI	Location	Server;
[C++]	public:	const	Oı	utputCache	Location	Server;
[VB]	Public	Const	Server	As	OutputCac	heLocation
[JScript]	public	var	Server	:	OutputCacl	heLocation;

# Description

The output cache is located on the Web server where the request was processed.

Page class (System.Web.UI)

**ToString** 

# Description

Represents an .aspx file requested from a server that hosts an ASP.NET Web application.

1	The	Page class is a	associated w	ith files that have an	ı .aspx ext	tension. These
2	files are co	ompiled at runtii	me as a <b>Pag</b> e	e object and cached i	n server m	emory.
3	ToS	String				
4						
5	[C#]	protected	const	string	postEven	ntArgumentID;
6	[C++]	protected:	const	String*	postEven	ntArgumentID;
7	[VB]	Protected	Const	postEventArgumen	tID A	As String
8	[JScript]	protected	var	postEventArgume	ntID	: String;
9						
10	Description	on				
11	To	String				
12						
13	[C#]	protected	const	string	postE	eventSourceID;
14	[C++]	protected:	cons	t String*	postE	EventSourceID;
15	[VB]	Protected	Const	postEventSource	ID A	As String
16	[JScript]	protected	var	postEventSource	eID	: String;
17						
18	Descriptio	on				
19	Pag	ge				
20		ample Syntax:				
21	То	String				
22						<b>.</b>
23	[C#]		_	oublic		Page();
24	[C++]			public:		Page();
25	VB]		Public	Sub		New()

yes pilc 509-324-9256 1216 *MS1-863US.APP* 

1	[JScript]	pι	ublic	func	etion	Page();
2						
3	Description					
4	Initiali	izes a new insta	ance of the Sy	stem.Web.U	J <b>I.Page</b> class.	
5	The de	efault construct	tor initializes	all fields to the	heir default valu	es.
6	Applic	cation				
7	ToStri	ng				
8						
9	[C#]	public	HttpApplicati	onState	Application	{get;}
10	[C++] pu	ıblic:pro	operty Htt	pApplication	State* get_A	Application();
11	[VB] Publi	ic ReadOnly	Property	Application	As HttpApp	olicationState
12	[JScript] p	oublic functi	ion get A	Application()	: HttpApp	licationState;
13						
14	Description					
15	Gets t	the <b>Applicatio</b>	n object for th	ne current We	eb request.	
16	AspC	ompatMode				
17	ToStr	ing				
18						
19	[C#]	bool		AspCompa		{set;}
20		protected:	property	void	set_AspCompa	
21	[VB]	Property	-	ompatMode	As	Boolean
22	[JScript]	protected	function	set	AspCompatMo	ode(Boolean);
23						
24	Description					
25						

Sets a value indicating whether the page can be executed on a single-threaded apartment (STA) thread.

When set to **true**, this allows the page to be executed on a single-threaded apartment (STA) thread. This allows the page to call STA components, such as components developed with Visual Basic 6.0. Setting this to **true** also allows the page to call COM+ 1.0 components that require access to the unmanaged ASP built-in objects. These are accessible through the ASP **ObjectContext** object or the **OnStartPage** method.

AutoHandlers

Buffer

**ToString** 

## Description

Sets a value indicating whether the page output is buffered.

In most circumstances, do not set this property in code. Set the **buffer** attribute to **true** using the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

Cache

**ToString** 

[C#]	publi	ic (	Cache	Cache		{get;}
[C++]	public:	proj	perty	Cache*	get	_Cache();
[VB]	Public	ReadOnly	Property	Cache	As	Cache
[JScript]	public	function	get	Cache()	:	Cache;

lee@hayes pilc 509-324-9256 1218 *MS1-863US.APP* 

1

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets the **System.Web.Caching.Cache** object associated with the application in which the page resides.

An application's **Cache** object allows you to store and retrieve arbitrary data on subsequent requests. The cache is not specifically associated with a page or user session. It is used primarily to enhance application performance. For more information, see .

ChildControlsCreated

ClientID

ClientTarget

**ToString** 

# Description

Gets or sets a value that allows you to override automatic detection of browser capabilities and to specify how a page renders for particular browser clients.

If you do not set this property, the **System.Web.HttpBrowserCapabilities** object associated with the **System.Web.UI.Page.Request** property reflects the capabilities of the client browser.

CodePage

**ToString** 

[C#] int CodePage {set;}

lee@haves pilc 509-324-9256 1219 MS1-863US.APP

[C++]	protected:	property	void	se	et_CodePage(int);
[VB]	Property	CodePage		As	Integer
[JScript]	protected	function	set		CodePage(int);

Sets the code page identifier for the current System. Web. UI. Page.

In most circumstances, do not set this property in code. Set the CodePage attribute to the value you want using the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

ContentType

**ToString** 

[C#]	string		Content?	Type	{set;}
[C++]	protected:	property	void	set_ContentTyp	pe(String*);
[VB]	Property	Conter	ntType	As	String
[JScript]	protected	function	set	ContentT	ype(String);

# Description

Sets the HTTP MIME type for the **System.Web.HttpResponse** object associated with the page.

In most circumstances, do not set this property in code. Set the ContentType attribute using the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

Context

**ToString** 

Context {get;} override HttpContext protected [C#] HttpContext\* get Context(); virtual protected: property [C++]Overrides Protected ReadOnly Property Context As HttpContext HttpContext; Context() protected function get [JScript]

## Description

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets the System. Web. HttpContext object associated with the page.

This property provides programmatic access to the context the page runs in, including information about the request, response, session and application.

Controls

Culture

**ToString** 

### Description

Sets the culture ID for the **System.Threading.Thread** object associated with the page.

In most circumstances, do not set this property in code. Set the **Culture** attribute in the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

EnableViewState

**ToString** 

[C#] public override bool EnableViewState {get; set;}

### Description

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets a value indicating whether the page maintains its view state, and the view state of any server controls it contains, when the current page request ends.

EnableViewStateMac

**ToString** 

EnableViewStateMac set;} {get; bool protected [C#] get EnableViewStateMac();protected: bool [C++]protected: property set EnableViewStateMac(bool); void property Boolean EnableViewStateMac As Protected Property [VB] [JScript] protected function get EnableViewStateMac() : Boolean;protected EnableViewStateMac(Boolean); function set

# Description

Gets or sets a value indicating whether ASP.NET should run a machine authentication check (MAC) on the page's view state when the page is posted back from the client.

In most circumstances, do not set this property in code. Set the EnableViewStateMac attribute to true using the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

ErrorPage

**ToString** 

[C#] public string ErrorPage {get; set;}
[C++] public: \_\_property String\* get\_ErrorPage();public: \_\_property void
set ErrorPage(String\*);

[VB] Public Property ErrorPage As String [JScript] public function get ErrorPage() : String; public function set ErrorPage(String);

# Description

Gets or sets the error page to which the requesting browser should be redirected in the event of an unhandled page exception.

**Events** 

FileDependencies

**ToString** 

# Description

Sets an array of files that the current **System.Web.HttpResponse** object is dependent upon.

In most circumstances, do not set this property in code. Set the **FileDependencies** attribute to **true** using the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

HasChildViewState

ID

**ToString** 

8

1

3

5

6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Description

Gets or sets an identifier for a particular instance of the System. Web.UI.Page class.

IsPostBack

**ToString** 

{get;} IsPostBack bool public [C#] get IsPostBack(); bool property public: [C++]Boolean As **Property** IsPostBack Public ReadOnly [VB] Boolean; IsPostBack() function [JScript] public get

Description

Gets a value indicating whether the page is being loaded in response to a client postback, or if it is being loaded and accessed for the first time.

IsReusable

**ToString** 

lee **@**hayes plic 509-324-9256 1224 *MS1-863US.APP* 

1									
2	[C#]	public		bool	IsReusabl	e	{get;}		
3	[C++]	public:	p	roperty	bool	get_Is	Reusable();		
4	[VB]	Public Re	eadOnly	Property	IsReusable	As	Boolean		
5	[JScript]	public	function	get	IsReusable()	:	Boolean;		
6									
7	Descripti	on							
8	Pa	ge class can be	e cached/reu	ised					
9	IsT	TrackingViewS	State						
10	Is*	Valid							
11	Тс	String							
12									
13									
14	Descripti	ion							
15					lidation succeed				
16	İ	For this property to return true, all validation server controls in the							
17	System.Web.UI.Page.Validators property must validate successfully.								
18	LCID								
19	ToString								
20							( , )		
21	[C#]		int		LCID		{set;}		
22	[C++]	protecte		property	void		t_LCID(int);		
23	[VB]	Prop		LCID		S	Integer		
24	[JScript]	pro	tected	function	n set		LCID(int);		
	11								

1

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Sets the locale identifier for the **System.Threading.Thread** object associated with the page.

In most circumstances, do not set this property in code. Set the **LCID** attribute in the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

NamingContainer

Page

Parent

Request

**ToString** 

# Description

Gets the System. Web. HttpRequest object for the requested page.

The **Request** object contains information about current incoming HTTP request.

Response

**ToString** 

{get;} HttpResponse Response public [C#] HttpResponse\* get Response(); public: property [C++]HttpResponse Public ReadOnly Property Response As [VB] HttpResponse; Response() [JScript] function public get

1

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets the **System.Web.HttpResponse** object associated with the **System.Web.UI.Page**. This object allows you to send HTTP response data to a client, and contains information about that response.

ResponseEncoding

**ToString** 

ResponseEncoding {set;} string [C#] set ResponseEncoding(String\*); void protected: property [C++]String As ResponseEncoding Property [VB] ResponseEncoding(String); function set protected [JScript]

# Description

Sets the encoding language for the current System.Web.HttpResponse object.

In most circumstances, do not set this property in code. Set the **ResponseEncoding** attribute to the value you want using the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

Server

**ToString** 

HttpServerUtility {get;} public Server [C#] HttpServerUtility\* get Server(); public: property [C++]HttpServerUtility **Property** Server As Public ReadOnly [VB]

[JScript] public function get Server() : HttpServerUtility;

Description

2

3

4

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets the Server object, which is an instance of the System. Web. HttpServerUtility class.

This property provides access to the frequently used System.Web.HttpServerUtility.HtmlEncode(System.String) and System.Web.HttpServerUtility.MapPath(System.String) methods, among others.

Session

**ToString** 

HttpSessionState Session {get;} virtual public [C#] get\_Session(); HttpSessionState\* virtual public: property [C++][VB] Overridable Public ReadOnly Property Session As HttpSessionState HttpSessionState; Session() function public get [JScript]

Description

Gets the current Session object provided by ASP.NET.

This property provides information about the current request's session. A Session object is maintained for each user that requests a page or document from an ASP.NET application. Variables stored in the Session object are not discarded when the user moves from page to page in the application; instead, these variables persist as long as the user is accessing pages in your application. For more information about session state, see .

16

17

18

19

20

21

22

23

24

Site SmartNavigation 2 **ToString** 3 5 Description Gets or sets a value indicating whether smart navigation is enabled. 7 In most circumstances, do not set this property in code. Set the 8 SmartNavigation attribute to true in the directive in the .aspx file. When the page 9 is requested, the dynamically generated class sets this property. 10 SupportAutoEvents 11 TemplateSourceDirectory 12 Trace 13 **ToString** 14

# Description

Gets the System. Web. TraceContext object for the current Web request.

Tracing tracks and presents the execution details about a Web request. For trace data to be visible in a rendered page, you must enable tracing at the page or application level.

TraceEnabled

**ToString** 

[C#] bool TraceEnabled {set;}

[C++] protected: \_\_property void set\_TraceEnabled(bool);
[VB] Property TraceEnabled As Boolean
[JScript] protected function set TraceEnabled(Boolean);

### Description

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Sets a value indicating whether tracing is enabled for the System. Web.UI.Page.

In most circumstances, do not set this property in code. Set the **Trace** attribute to **true** in the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

TraceModeValue

**ToString** 

{set;} TraceModeValue TraceMode [C#] set\_TraceModeValue(TraceMode); void protected: property [C++]TraceMode As TraceModeValue Property [VB] TraceModeValue(TraceMode); function [JScript] protected set

### Description

Sets the mode in which trace statements are displayed on the page.

In most circumstances, do not set this property in code. Set the **TraceMode** attribute in the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

TransactionMode

**ToString** 

[C#]	int		TransactionMode			
[C++]	protected:	property	void	set_Transac	tionMode(int);	
[VB]	Property	Transac	ctionMode	As	Integer	
[JScript]	protected	function	set	Transac	tionMode(int);	

Sets the level of transaction support for the page.

In most circumstances, do not set this property in code. Set the **Transaction** attribute in the directive in the .aspx file. When the page is requested, the dynamically generated class sets the property.

UICulture

**ToString** 

[C#]	string	UICulture	e	{set;}	
[C++]	protected:	property	void	set_UICu	ılture(String*);
[VB]	Property	UICul	lture	As	String
[JScript]	protected	function	set	UIC	Culture(String);

# Description

Sets the UI ID for the **System.Threading.Thread** object associated with the page.

This property is a shortcut for **System.Threading.Thread.CurrentThread**. The culture is a property of the executing thread

UniqueID

24

25

User

2

**ToString** 

Description Gets information about the user making the page request. An IPrincipal object represents the security context of the user on whose behalf the code is running, including that user's identity and any roles to which they belong. Validators **ToString** 11 12 public ValidatorCollection Validators {get;} [C#] 13 get\_Validators(); ValidatorCollection\* public: property [C++]ValidatorCollection [VB] Public ReadOnly Property Validators As 15 ValidatorCollection; [JScript] public function get Validators() 17 Description 18 Gets a collection of all validation controls contained on the requested page. 19 ViewState 20 ViewStateIgnoresCase 21 Visible 22 **ToString** 

2

4

5

6

7

8

10

11

13

15

17

18

19

20

21

22

23

24

25

Gets or sets a value indicating whether the **System.Web.UI.Page** object should be rendered.

AspCompatBeginProcessRequest

IAsyncResult AspCompatBeginProcessRequest(HttpContext [C#] protected cb, object context, AsyncCallback extraData); [C++] protected: IAsyncResult\* AspCompatBeginProcessRequest(HttpContext\* AsyncCallback\* Object\* cb, extraData); context, [VB] Protected Function AspCompatBeginProcessRequest(ByVal context As HttpContext, ByVal cb As AsyncCallback, ByVal extraData As Object) As **IAsyncResult** 

[JScript] protected function AspCompatBeginProcessRequest(context : HttpContext, cb : AsyncCallback, extraData : Object) : IAsyncResult;

# Description

Initiates a request for Active Server Page (ASP) resources. This method is provided for compatibility with legacy ASP applications.

Do not call this method. An **System.Web.HttpContext** object with information about the current request. The callback method. Any extra data needed to process the request in the same manner as an ASP request.

Asp Compat End Process Request

1								
2	[C#] protected void AspCompatEndProcessRequest(IAsyncResult result);							
3	[C++] protected: void AspCompatEndProcessRequest(IAsyncResult* result);							
4	[VB] Protected Sub AspCompatEndProcessRequest(ByVal result As							
5	IAsyncResult)							
6	[JScript] protected function AspCompatEndProcessRequest(result :							
7	IAsyncResult);							
8								
9	Description							
10	Terminates a request for Active Server Page (ASP) resources. This method							
11	is provided for compatibility with legacy ASP applications.							
12	Do not call this method. The ASP page generated by the request.							
13	CreateHtmlTextWriter							
14								
15	[C#] protected virtual HtmlTextWriter CreateHtmlTextWriter(TextWriter tw);							
16	[C++] protected: virtual HtmlTextWriter* CreateHtmlTextWriter(TextWriter*							
17	tw);							
18	[VB] Overridable Protected Function CreateHtmlTextWriter(ByVal tw As							
19	TextWriter) As HtmlTextWriter							
20	[JScript] protected function CreateHtmlTextWriter(tw : TextWriter) :							
21	HtmlTextWriter;							
22								
23	Description							
24	Creates an System.Web.UI.HtmlTextWriter object to render the page's							
25	content.							

5

7

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Return Value: The specified System.Web.UI.HtmlTextWriter or System.Web.UI.Html32TextWriter object.

The based information object created is from the on System. Web. HttpBrowserCapabilities object associated with the page request. this System.Web.UI.HtmlTextWriter Usually, object is an System.Web.UI.Html32TextWriter object. For derived pages, you can override this method writer Creates to create a custom text an System. Web. UI. Html Text Writer object to render the page's content. If the IsUplevel property is set to false, an Html32TextWriter object is created to render requests originating from downlevel browsers. For derived pages, you can override this method to create a custom text writer. The text writer to create.

### DesignerInitialize

[C#]	public	void	DesignerInitialize();
[C++]	public:	void	DesignerInitialize();
[VB]	Public	Sub	DesignerInitialize()
[JScript]	public	function	DesignerInitialize();

### Description

Performs any initialization of the instance of the **System.Web.UI.Page** class that is required by RAD designers. This method is only used at design time.

#### DeterminePostBackMode

[C#] protected virtual NameValueCollection DeterminePostBackMode(); [C++] protected: virtual NameValueCollection\* DeterminePostBackMode(); [VB] Overridable Protected Function DeterminePostBackMode() As NameValueCollection

[JScript] protected function DeterminePostBackMode(): NameValueCollection;

#### Description

Determines the type of request made for the **Page** class. This information is based on whether the page was posted back, and whether the GET or POST HTTP method was used for the request. *Return Value:* If the postback used the POST method, the Form information is returned from the **Context** object. If the postback used the GET method, the query string information is returned. If the page is being requested for the first time, **null** is returned.

#### GetPostBackClientEvent

[C#] public string GetPostBackClientEvent(Control control, string argument); [C++] public: String\* GetPostBackClientEvent(Control\* control, String\* argument);

[VB] Public Function GetPostBackClientEvent(ByVal control As Control, ByVal argument As String) As String

[JScript] public function GetPostBackClientEvent(control : Control, argument : String) : String;

#### Description

Obtains a reference to a client-side script function that causes, when invoked, a server postback to the form.

5

6

8

10

11

13

14

15

16

17

18

20

21

22

23

24

Return Value: The String that represents the client event. The server control that receives the client event postback. A System.String argument that is passed to the System.Web.UI.IPostBackEventHandler.RaisePostBackEvent(System.String) method.

### GetPostBackClientHyperlink

[C#] public string GetPostBackClientHyperlink(Control control, string argument); [C++] public: String\* GetPostBackClientHyperlink(Control\* control, String\* argument);

[VB] Public Function GetPostBackClientHyperlink(ByVal control As Control, ByVal argument As String) As String

[JScript] public function GetPostBackClientHyperlink(control : Control, argument

: String) : String;

#### Description

javascript: to the beginning of the return System.Web.UI.Page.GetPostBackEventReference(System.Web.UI.Control) allow call hyperlink post back processing the on Return Value: The name of the client-side function and the ID of the server control that processed the function and argument passed to the control. The server control to process the postback. The parameter passed to the server control.

#### GetPostBackEventReference

[C#] public string GetPostBackEventReference(Control control);
[C++] public: String\* GetPostBackEventReference(Control\* control);

[VB] Publ	ic Function	GetPostBack	EventReferen	ce(ByVal co	ntrol As	Control) A
String						

[JScript] public function GetPostBackEventReference(control : Control) : String;
Obtains a reference to a client-side script function that causes, when invoked, the
server to postback to the page.

#### Description

Obtains a reference to a client-side script function that causes, when invoked, the server to postback to the page.

Return Value: The text of the client-side function call that can be inserted into a client-side event handler. The server control to process the postback on the server.

#### GetPostBackEventReference

[C#] public string GetPostBackEventReference(Control control, string argument); [C++] public: String\* GetPostBackEventReference(Control\* control, String\* argument);

[VB] Public Function GetPostBackEventReference(ByVal control As Control,
ByVal argument As String) As String

[JScript] public function GetPostBackEventReference(control : Control, argument

: String) : String;

#### Description

Obtains a reference to a client-side script function that causes, when invoked, the server to postback to the page. It also passes a parameter to the server control that performs the postback processing on the server.

3

4

5

10

11

12

13

15

16

18

20

21

22

23

24

Return Value: The text of the client-side function call that can be inserted into a client-side event handler. The server control to process the postback. The parameter passed to the server control.

### GetTypeHashCode

[C#] public virtual int GetTypeHashCode(); [C++]public: virtual int GetTypeHashCode(); Overridable [VB] Public Function GetTypeHashCode() As Integer [JScript] function GetTypeHashCode() public int;

#### Description

Retrieves a hash code that is generated by **Page** objects that are generated at run time. This hash code is unique to the **Page** object's control hierarchy. Return Value: The hash code generated at run time. The default is 0.

#### InitOutputCache

[C#] protected virtual void InitOutputCache(int duration, string varyByHeader, string varyByCustom, OutputCacheLocation location, string varyByParam); [C++] protected: virtual void InitOutputCache(int duration, String\* varyByHeader, String\* varyByCustom, OutputCacheLocation location, String\* varyByParam);

[VB] Overridable Protected Sub InitOutputCache(ByVal duration As Integer, ByVal varyByHeader As String, ByVal varyByCustom As String, ByVal location As OutputCacheLocation, ByVal varyByParam As String) [JScript] protected function InitOutputCache(duration: int, varyByHeader: String,

lee@hayes pilc 509-324-9256 1239 MS1-863US.APP

varyByCustom: String, location: OutputCacheLocation, varyByParam: String);

Description

Initializes the output cache for the current page request.

You should not call this method. To enable and manipulate output caching for a page, use either the directive in the .aspx file, or the methods and properties of the System.Web.HttpCachePolicy class. The latter are accessible through Response.Cache syntax in the page's code-declaration block or code-behind file. For more information, see . An integer representing the amount of time that objects stored in the output cache are valid. A semi-colon separated list of headers that content from the output cache will vary by. A string that represents the Vary HTTP header. The location of the output cache as specified by the System.Web.UI.OutputCacheLocation enumeration. A semi-colon separated list of parameters, received by a GET or POST method, that content from the output cache will vary by.

# **IsClientScriptBlockRegistered**

[C#] public bool IsClientScriptBlockRegistered(string key);
[C++] public: bool IsClientScriptBlockRegistered(String\* key);
[VB] Public Function IsClientScriptBlockRegistered(ByVal key As String) As
Boolean
[JScript] public function IsClientScriptBlockRegistered(key: String): Boolean;

Description

Determines if the client script block is registered with the page.

Return Value: Returns true if the script block is registered; otherwise, false.

System.Web.UI.Page.RegisterClientScriptBlock(System.String,System.String) to avoid unnecessarily assembling the client-side script. This particularly important if the script requires a large amount of server resources to create. The string key of the client script to search for.

**IsStartupScriptRegistered** 

[C#] public bool IsStartupScriptRegistered(string key);
[C++] public: bool IsStartupScriptRegistered(String\* key);
[VB] Public Function IsStartupScriptRegistered(ByVal key As String) As Boolean
[JScript] public function IsStartupScriptRegistered(key : String) : Boolean;

## Description

Determines if the client startup script is registered with the System.Web.UI.Page object.

Return Value: true if the startup script is registered; otherwise, false.

Call this method before calling System.Web.UI.Page.RegisterStartupScript(System.String,System.String) to avoid unnecessarily assembling the client-side script. This particularly important if the script requires a large amount of server resources to create. The string key of the startup script to search for.

LoadPageStateFromPersistenceMedium

10

11

12

13

14

16

17

18

20

22

23

24

25

[C#]	protected	virtual	object	LoadPageStateFromPersistenceMedium();
[C++]	protected:	virtual	Object*	Load Page State From Persistence Medium ();
[VB] (	Overridable	Protected	Function	LoadPageStateFromPersistenceMedium()
As				Object
[JScrip	t] protected	function	LoadPage	StateFromPersistenceMedium(): Object;
Descrip	otion			
]	Loads any	saved vie	ew-state in	nformation to the System.Web.UI.Page
object.	Override th	is method	if you wa	nt to load the Page view state in anything

hidden field. than other a

If you want to specify something other than hidden fields to save view state this method, also override the when using you must SaveStateToPersistenceMedium method.

## MapPath

Return Value: The saved view state.

[C#] public MapPath(string virtualPath); string String\* MapPath(String\* [C++]public: virtualPath); [VB] Public Function MapPath(ByVal virtualPath As String) As String function MapPath(virtualPath [JScript] public String) String;

## Description

Retrieves the physical path that a virtual path, either absolute or relative, maps to.

Return Value: The physical path associated with the virtual path. A System.String 1 that represents the virtual path. 2 ProcessRequest 3 [C#] public ProcessRequest(HttpContext context); void 5 sealed void ProcessRequest(HttpContext\* [C++]public: context); [VB] NotOverridable Public Sub ProcessRequest(ByVal context As HttpContext) [JScript] public function ProcessRequest(context HttpContext); 8 9 Description 10 RaisePostBackEvent 11 12 [C#] protected virtual void RaisePostBackEvent(IPostBackEventHandler 13 sourceControl, string eventArgument); 14 [C++] protected: virtual void RaisePostBackEvent(IPostBackEventHandler\* 15 String\* sourceControl, eventArgument); 16 [VB] Overridable Protected Sub RaisePostBackEvent(ByVal sourceControl As 17 IPostBackEventHandler, ByVal As String) eventArgument 18 protected function RaisePostBackEvent(sourceControl [JScript] 19 IPostBackEventHandler, String); eventArgument 20 21 Description 22 Notifies the server control that caused postback that it should handle an 23 incoming postback event. 24

The Page calls this when a post back occurs. This occurs in the page lifecycle after loading and change notification have completed, but before prerendering occurs. The ASP.NET server control that caused postback. This control must implement the IPostBackEventHandler interface. The post back argument.

## RegisterArrayDeclaration

[C#] public void RegisterArrayDeclaration(string arrayName, string arrayValue); [C++] public: void RegisterArrayDeclaration(String\* arrayName, String\* arrayValue);

[VB] Public Sub RegisterArrayDeclaration(ByVal arrayName As String, ByVal arrayValue As String)

[JScript] public function RegisterArrayDeclaration(arrayName : String, arrayValue : String);

### Description

Declares a value that will be declared as an ECMAScript array declaration when the page renders. This can be used by script-based controls to declare themselves within an array so that a client script library can work with all the controls of the same type. The name of the array in which to declare the value. The value to place in the array.

## RegisterClientScriptBlock

[C#] public virtual void RegisterClientScriptBlock(string key, string script); [C++] public: virtual void RegisterClientScriptBlock(String\* key, String\* script);

[VB] Overridable Public Sub RegisterClientScriptBlock(ByVal key As String, ByVal script As 2 [JScript] public function RegisterClientScriptBlock(key: String, script: String); 4 Description 5 Allows ASP.NET server controls to emit client-side script blocks into the 6 System.Web.UI.Page. 7 The client-side script is emitted jut after the opening tag of the Page 8 object's 9 Top of Form 10 element. The script block is emitted as you define to the output stream, so 11 you must include both tags of the 12 Bottom of Form 13 14 15 16 System.Web.UI.Design 17 18 Description 19 The System. Web. UI. Design namespace contains classes that can be used 20 to extend design-time support for Web Forms. 21 CalendarDataBindingHandler class (System.Web.UI.Design) 22 23

String)

Description

1	Provides a data binding handler for calendar data.
2	Constructors:
3	CalendarDataBindingHandler
4	Example Syntax:
5	
6	[C#] public CalendarDataBindingHandler();
7	[C++] public: CalendarDataBindingHandler();
8	[VB] Public Sub New()
9	[JScript] public function CalendarDataBindingHandler();
10	Methods:
11	DataBindControl
12	
13	[C#] public override void DataBindControl(IDesignerHost designerHost, Control
14	control);
15	[C++] public: void DataBindControl(IDesignerHost* designerHost, Control*
16	control);
17	[VB] Overrides Public Sub DataBindControl(ByVal designerHost As
18	IDesignerHost, ByVal control As Control)
19	[JScript] public override function DataBindControl(designerHost: IDesignerHost
20	control: Control);
21	
22	Description
23	Adds this data binding to the specified control. The designer host for the
24	document that contains the control. The control to add this data binding to.
25	ColorBuilder class (System.Web.UI.Design)

1	ToString
2	
3	
4	Description
5	Launches a color editor that allows a user to select a color.
6	The
7	System. We b. UI. Design. Color Builder. Build Color (System. Component Model. In the color builder) and the color builder of the col
8	Component, System. Windows. Forms. Control, System. String) method launches
9	a user interface for selecting a color value.
10	BuildColor
11	
12	[C#] public static string BuildColor(IComponent component, Control owner,
13	string initialColor);
14	[C++] public: static String* BuildColor(IComponent* component, Control*
15	owner, String* initialColor);
16	[VB] Public Shared Function BuildColor(ByVal component As IComponent,
17	ByVal owner As Control, ByVal initialColor As String) As String
18	[JScript] public static function BuildColor(component : IComponent, owner :
19	Control, initialColor: String): String;
20	
21	Description
22	Launches a color editor to build a color.
23	Return Value: The color value, represented as a string in an HTML color format,

or null if the builder service could not be retrieved.

MS1-863US.APP

2

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

The returned string, if other than **null**, indicates a color in a valid HTML color format. Valid formats include named colors and color codes in RGB format (#RRGGBB). The component whose site is to be used to access design-time services. The control used to parent the picker window. The initial color to be shown in the picker window.

ControlDesigner class (System.Web.UI.Design)

**ToString** 

## Description

Provides a base class for Web server control designers.

ControlDesigner

Example Syntax:

**ToString** 

[C#] public ControlDesigner();

[C++] public: ControlDesigner();

[VB] Public Sub New()

[JScript] public function ControlDesigner();

Properties:

AllowResize

**ToString** 

[C#] public virtual bool AllowResize {get;}

[C++] public: property virtual bool get AllowResize();

1	[VB] Overridable Public ReadOnly Property AllowResize As Boolean
2	[JScript] public function get AllowResize(): Boolean;
3	
4	Description
5	Gets a value indicating whether or not the control can be resized.
6	AssociatedComponents
7	Behavior
8	Component
9	DataBindings
10	DesignTimeElement
11	DesignTimeElementView
12	ToString
13	
14	
15	Description
16	Gets the designer view control object for the designer.
17	DesignTimeHtmlRequiresLoadComplete
18	ToString
19	
20	[C#] public virtual bool DesignTimeHtmlRequiresLoadComplete {get;}
21	[C++] public:property virtual bool
22	get_DesignTimeHtmlRequiresLoadComplete();
23	[VB] Overridable Public ReadOnly Property
24	DesignTimeHtmlRequiresLoadComplete As Boolean
25	[JScript] public function get DesignTimeHtmlRequiresLoadComplete(): Boolean;

20

21

22

23

24

25

1	
2	Description
3	Gets a value indicating whether the designer must have completed loading
4	before the design time HTML can be used.
5	The default implementation returns false.
6	ID
7	ToString
8	
9	[C#] public virtual string ID {get; set;}
10	[C++] public:property virtual String* get_ID();public:property virtual void
11	set_ID(String*);
12	[VB] Overridable Public Property ID As String
13	[JScript] public function get ID(): String; public function set ID(String);
14	
15	Description
16	Gets or sets the ID for the control designer.
17	InheritanceAttribute
18	Inherited

# Description

IsDirty

**ToString** 

Gets or sets a value indicating whether the Web server control has been marked as changed.

1	ReadOnly
2	ToString
3	
4	[C#] public bool ReadOnly {get; set;}
5	[C++] public:property bool get_ReadOnly();public:property void
6	set_ReadOnly(bool);
7	[VB] Public Property ReadOnly As Boolean
8	[JScript] public function get ReadOnly(): Boolean; public function set
9	ReadOnly(Boolean);
10	
11	Description
12	Gets or sets a value indicating whether the control's associated design
13	surface is set to read-only.
14	ShadowProperties
15	ShouldCodeSerialize
16	Verbs
17	CreatePlaceHolderDesignTimeHtml
18	
19	[C#] protected string CreatePlaceHolderDesignTimeHtml();
20	[C++] protected: String* CreatePlaceHolderDesignTimeHtml();
21	[VB] Protected Function CreatePlaceHolderDesignTimeHtml() As String
22	[JScript] protected function CreatePlaceHolderDesignTimeHtml(): String; Creates
23	a simple HTML section for the control that can be used to display information
24	related to the control at design-time when there is not enough information to
25	display a meaningful representation of the control.

### Description

Creates a simple HTML section for the control that can be used to display information related to the control at design-time when there is not enough information to display a meaningful representation of the control.

Return Value: A string containing place-holder design-time HTML.

This method returns a string containing HTML in a standard place-holder style that an empty control can display at design-time to provide some basic information about the control. The string that is returned contains the type of the control and its ID.

CreatePlaceHolderDesignTimeHtml

[C#] protected string CreatePlaceHolderDesignTimeHtml(string instruction);
[C++] protected: String\* CreatePlaceHolderDesignTimeHtml(String\* instruction);
[VB] Protected Function CreatePlaceHolderDesignTimeHtml(ByVal instruction
As String) As String

 $[JScript]\ protected\ function\ Create Place Holder Design Time Html (instruction: Create Place Holder Design Time Html)$ 

String): String;

### Description

Creates a simple HTML section for the control, using the specified additional information, that can be used to display information related to the control at design-time when there is not enough information to display a meaningful representation of the control.

lee@hayes plic 509-324-9256

Return Value: A string containing place holder design-time HTML and the specified string. 2 This method returns a string containing HTML in a standard place-holder 3 style that an empty control can display at design-time to provide some basic information about the control. The string that is returned contains the type of the 5 control and its ID. A string conatining information to add to the HTML section. **GetDesignTimeHtml** 7 8 [C#] public virtual string GetDesignTimeHtml(); [C++] public: virtual String\* GetDesignTimeHtml(); 10 [VB] Overridable Public Function GetDesignTimeHtml() As String 11 [JScript] public function GetDesignTimeHtml(): String; 12 13 Description 14 Gets the HTML to be used for the design-time representation of the control. 15 Return Value: The design-time HTML for the control. 16 **GetEmptyDesignTimeHtml** 17 18 [C#] protected virtual string GetEmptyDesignTimeHtml(); [C++] protected: virtual String\* GetEmptyDesignTimeHtml(); 20 [VB] Overridable Protected Function GetEmptyDesignTimeHtml() As String 21 [JScript] protected function GetEmptyDesignTimeHtml(): String; 22 23

Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets the HTML to be used for the design-time representation of a blank control.

Return Value: The HTML used for the design-time representation of a blank control. This is the name of the component, by default.

The default behavior is to return a string containing the name of the component. This method is called by

**System.Web.UI.Design.ControlDesigner.GetDesignTimeHtml** when there is no design-time HTML.

GetErrorDesignTimeHtml

[C#] protected virtual string GetErrorDesignTimeHtml(Exception e);

[C++] protected: virtual String\* GetErrorDesignTimeHtml(Exception\* e);

[VB] Overridable Protected Function GetErrorDesignTimeHtml(ByVal e As

Exception) As String

[JScript] protected function GetErrorDesignTimeHtml(e: Exception): String;

Description

Gets the HTML to be used for the design-time representation of the control after an error has been encountered.

Return Value: The HTML for the specified exception. The exception that occurred.

GetPersistInnerHtml

[C#] public virtual string GetPersistInnerHtml();

[C++] public: virtual String\* GetPersistInnerHtml();

1	[VB] Overridable Public Function GetPersistInnerHtml() As String
2	[JScript] public function GetPersistInnerHtml(): String;
3	
4	Description
5	Gets the persistable inner HTML.
6	Return Value: The persistable inner HTML.
7	Initialize
8	
9	[C#] public override void Initialize(IComponent component);
10	[C++] public: void Initialize(IComponent* component);
11	[VB] Overrides Public Sub Initialize(ByVal component As IComponent)
12	[JScript] public override function Initialize(component : IComponent);
13	
14	Description
15	Initializes the designer with the specified component.
16	This method is called by the designer host to establish the component being
17	designed. The control element being designed.
18	IsPropertyBound
19	
20	[C#] public bool IsPropertyBound(string propName);
21	[C++] public: bool IsPropertyBound(String* propName);
22	[VB] Public Function IsPropertyBound(ByVal propName As String) As Boolean
23	[JScript] public function IsPropertyBound(propName : String) : Boolean;
24	
25	Description
	·

2

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a value indicating whether the specified property is data-bound. Return Value: true if the property is data bound; othwerise, false. The property to check for data binding. **OnBehaviorAttached** [C#] protected override void OnBehaviorAttached(); [C++] protected: void OnBehaviorAttached(); [VB] Overrides Protected Sub OnBehaviorAttached() [JScript] protected override function OnBehaviorAttached(); Description Called when the designer is attached to the behavior. **OnBindingsCollectionChanged** [C#] protected override void OnBindingsCollectionChanged(string propName); [C++] protected: void OnBindingsCollectionChanged(String\* propName); [VB] Overrides Protected Sub OnBindingsCollectionChanged(ByVal propName As String) [JScript] protected override function OnBindingsCollectionChanged(propName: String); Description Called when the data bindings collection changes.

1 This method is called when the bindings collection has been changed by an external caller. In Visual Studio.NET, this method is called by the 2 WebFormDataTab. The property to test for changes in its bindings collection. 3 **OnComponentChanged** 5 [C#] public virtual void OnComponentChanged(object sender, ComponentChangedEventArgs ce); [C++] public: virtual void OnComponentChanged(Object\* sender, ComponentChangedEventArgs\* ce); [VB] Overridable Public Sub OnComponentChanged(ByVal sender As Object, 10 ByVal ce As ComponentChangedEventArgs) 11 [JScript] public function OnComponentChanged(sender : Object, ce : 12 ComponentChangedEventArgs); 13 14 Description 15 Called when the component changes. 16 This method is called when a property is changed. It allows the 17 implementer to do any processing that may be needed after a property change. The 18 object that is the source of the event. A 19 System.ComponentModel.Design.ComponentChangedEventArgs object that 20 provides data about the event. 21 **OnControlResize** 22 23 [C#] protected virtual void OnControlResize(); 24 [C++] protected: virtual void OnControlResize();

1	[VB] Overridable Protected Sub OnControlResize()
2	[JScript] protected function OnControlResize();
3	
4	Description
5	Called when the design-time control has been resized.
6	This method is typically only called by the design-time environment when
7	a user action causes the control to be resized. This method may be called several
8	times during a resize process to display the updated size of the control before the
9	resize process is completed. The width and height properties of the control are
10	updated before this method is called.
11	PreFilterProperties
12	
13	[C#] protected override void PreFilterProperties(IDictionary properties);
14	[C++] protected: void PreFilterProperties(IDictionary* properties);
15	[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As
16	IDictionary)
17	[JScript] protected override function PreFilterProperties(properties : IDictionary)
18	RaiseResizeEvent
19	
20	[C#] public void RaiseResizeEvent();
21	[C++] public: void RaiseResizeEvent();
22	[VB] Public Sub RaiseResizeEvent()
23	[JScript] public function RaiseResizeEvent();
24	
25	Description

1	Raises the System.Web.UI.Design.ControlDesigner.OnControlResize
2	event.
3	UpdateDesignTimeHtml
4	
5	[C#] public virtual void UpdateDesignTimeHtml();
6	[C++] public: virtual void UpdateDesignTimeHtml();
7	[VB] Overridable Public Sub UpdateDesignTimeHtml()
8	[JScript] public function UpdateDesignTimeHtml();
9	
10	Description
11	Updates the design-time HTML.
12	ControlParser class (System.Web.UI.Design)
13	UpdateDesignTimeHtml
14	
15	
16	Description
17	Provides methods for parsing the code for a control.
18	ParseControl
19	
20	[C#] public static Control ParseControl(IDesignerHost designerHost, string
21	controlText);
22	[C++] public: static Control* ParseControl(IDesignerHost* designerHost, String*
23	controlText);
24	[VB] Public Shared Function ParseControl(ByVal designerHost As
25	IDesignerHost, ByVal controlText As String) As Control

[JScript] public static function ParseControl(designerHost: IDesignerHost, controlText: String): Control; Initializes a new instance of the System.Web.UI.Design.ControlParser class using the specified designer host and control text.

### Description

Initializes a new instance of the **System.Web.UI.Design.ControlParser** class using the specified designer host and control text.

Return Value: The Control that the specified text represents, or **null** if the parser could not build the control. The designer host for the document. The text of the code for the control.

#### **ParseControl**

[C#] public static Control ParseControl(IDesignerHost designerHost, string controlText, string directives);

[C++] public: static Control\* ParseControl(IDesignerHost\* designerHost, String\* controlText, String\* directives);

[VB] Public Shared Function ParseControl(ByVal designerHost As IDesignerHost, ByVal controlText As String, ByVal directives As String) As Control

[JScript] public static function ParseControl(designerHost: IDesignerHost, controlText: String, directives: String): Control;

## Description

Initializes a new instance of the **System.Web.UI.Design.ControlParser** class using the specified designer host, control text and directives.

Return Value: The Control that the specified text represents, or **null** if the parser could not build the control. The designer host of the document. The text of the code for the control. The directives to include in the code for the control.

ParseTemplate

[C#] public static ITemplate ParseTemplate(IDesignerHost designerHost, string templateText);

[C++] public: static ITemplate\* ParseTemplate(IDesignerHost\* designerHost, String\* templateText);

[VB] Public Shared Function ParseTemplate(ByVal designerHost As IDesignerHost, ByVal templateText As String) As ITemplate
[JScript] public static function ParseTemplate(designerHost: IDesignerHost, templateText: String): ITemplate; Parses the specified text of a template.

### Description

Parses the specified text of a template.

Return Value: A new template based on the specified text. The designer host of the document. The text of the code for a template.

ParseTemplate

[C#] public static ITemplate ParseTemplate(IDesignerHost designerHost, string templateText, string directives);

[C++] public: static ITemplate\* ParseTemplate(IDesignerHost\* designerHost,

1	String* templateText, String* directives);
2	[VB] Public Shared Function ParseTemplate(ByVal designerHost As
3	IDesignerHost, ByVal templateText As String, ByVal directives As String) As
4	ITemplate
5	[JScript] public static function ParseTemplate(designerHost: IDesignerHost,
6	templateText: String, directives: String): ITemplate;
7	
8	Description
9	Parses the specified text of a template.
10	Return Value: A new template based on the specified text. The designer host of
11	the document. The text of the code for a template. Any directives to add to the
12	beginning of the code for the template.
13	ControlPersister class (System.Web.UI.Design)
14	ToString
15	
16	
17	Description
18	Provides helper functions for persisting Web server controls.
19	PersistControl
20	
21	[C#] public static string PersistControl(Control control);
22	[C++] public: static String* PersistControl(Control* control);
23	[VB] Public Shared Function PersistControl(ByVal control As Control) As String
24	[JScript] public static function PersistControl(control: Control): String; Gets a
25	string of persistence data that can persist a control.

	1
Descript	2
G	3
Return V	4
The Syst	5
Pe	6
	7
[C#] pub	8
[C++] pu	9
host);	10
[VB] Pul	11
host As I	12
[JScript]	13
IDesigne	14
	15
Descript	16
G	17
Return V	18
The Syst	19
Pe	20
	21
[C#] pub	22

$\boldsymbol{r}$	• ,•	
1)	escription	
~	sci ipiion	

ets a string of persistence data that can persist a control.

*'alue:* A string that contains the information to persist about the control.

tem.Web.UI.Control to persist.

ersistControl

lic static string PersistControl(Control control, IDesignerHost host);

ublic: static String\* PersistControl(Control\* control, IDesignerHost\*

blic Shared Function PersistControl(ByVal control As Control, ByVal

DesignerHost) As String

public static function PersistControl(control: Control, host:

erHost) : String;

### ion

ets a string of persistence data that can persist a control.

*'alue:* A string that contains the information to persist about the control.

tem.Web.UI.Control to persist. The designer host for the control.

ersistControl

[C#] public static void PersistControl(TextWriter sw, Control control);

[C++] public: static void PersistControl(TextWriter\* sw, Control\* control);

[VB] Public Shared Sub PersistControl(ByVal sw As TextWriter, ByVal control

As Control)

	1	[JScript] public static function PersistControl(sw: TextWriter, control: Control);
	2	
	3	Description
	4	Persists a control using the specified text writer. The
	5	System.IO.TextWriter to use. The System.Web.UI.Control to persist.
	6	PersistControl
	7	
	8	[C#] public static void PersistControl(TextWriter sw, Control control,
	9	IDesignerHost host);
and that	10	[C++] public: static void PersistControl(TextWriter* sw, Control* control,
	11	IDesignerHost* host);
	12	[VB] Public Shared Sub PersistControl(ByVal sw As TextWriter, ByVal control
	13	As Control, ByVal host As IDesignerHost)
	14	[JScript] public static function PersistControl(sw: TextWriter, control: Control,
	15	host: IDesignerHost);
	16	
	17	Description
	18	Persists a control using the specified text writer. The
	19	System.IO.TextWriter to use. The System.Web.UI.Control to persist. The
	20	designer host for the control.
	21	PersistInnerProperties
	22	
	23	[C#] public static string PersistInnerProperties(object component, IDesignerHost
	24	host);
	25	[C++] public: static String* PersistInnerProperties(Object* component,

Description

IDesignerHost\* host);

[VB] Public Shared Function PersistInnerProperties(ByVal component As Object, 2 ByVal host As IDesignerHost) As String 3 [JScript] public static function PersistInnerProperties(component : Object, host : IDesignerHost): String; Gets a string of persistance data that can persist the inner 5 properties of a control. 6 7 Description 8 Gets a string of persistance data that can persist the inner properties of a control. 10 Return Value: A string that contains the information to persist about the inner 11 properties of the control. The System. Web. UI. Control to persist. The designer 12 host of the control. 13 **PersistInnerProperties** 14 15 [C#] public static void PersistInnerProperties(TextWriter sw, object component, 16 IDesignerHost host); 17 [C++] public: static void PersistInnerProperties(TextWriter\* sw, Object\* 18 component, IDesignerHost\* host); 19 [VB] Public Shared Sub PersistInnerProperties(ByVal sw As TextWriter, ByVal 20 component As Object, ByVal host As IDesignerHost) 21 [JScript] public static function PersistInnerProperties(sw: TextWriter, component 22 : Object, host : IDesignerHost); 23

lee@hayes piik 509-324-9256 1265 *MS1-863US.APP* 

	1	Persists the inner properties of the control using the specified text writer.
	2	The System.IO.TextWriter to use. The System.Web.UI.Control to persist. The
	3	designer host of the control.
	4	DataBindingCollectionConverter class (System.Web.UI.Design)
	5	ToString
	6	
	7	
	8	Description
	9	Provides a type converter to convert data binding collections to various
	10	other representations.
	11	DataBindingCollectionConverter
	12	Example Syntax:
	13	ToString
	14	
Sant Anti He din, But	15	[C#] public DataBindingCollectionConverter();
	16	[C++] public: DataBindingCollectionConverter();
••	17	[VB] Public Sub New()
	18	[JScript] public function DataBindingCollectionConverter();
	19	ConvertTo
	20	
	21	[C#] public override object ConvertTo(ITypeDescriptorContext context,
	22	CultureInfo culture, object value, Type destinationType);
	23	[C++] public: Object* ConvertTo(ITypeDescriptorContext* context, CultureInfo*
	24	culture, Object* value, Type* destinationType);
	25	[VB] Overrides Public Function ConvertTo(ByVal context As

1	ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object,
2	ByVal destinationType As Type) As Object
3	[JScript] public override function ConvertTo(context : ITypeDescriptorContext,
4	culture : CultureInfo, value : Object, destinationType : Type) : Object;
5	
6	Description
7	Converts a data binding collection to the specified type.
8	Return Value: The object produced by the type conversion. An
9	System.ComponentModel.ITypeDescriptorContext that indicates the
10	component or control the data binding collection belongs to. A
11	System.Globalization.CultureInfo that can be used to provide additional culture
12	information. The object to convert. The type to convert to.
13	DataBindingCollectionEditor class (System.Web.UI.Design)
14	ToString
15	
16	
17	Description
18	Provides user interface for editing a collection of data bindings.
19	DataBindingCollectionEditor
20	Example Syntax:
21	ToString
22	
23	[C#] public DataBindingCollectionEditor();
24	[C++] public: DataBindingCollectionEditor();
25	

1	[VB] Public Sub New()
2	[JScript] public function DataBindingCollectionEditor();
3	EditValue
4	
5	[C#] public override object EditValue(ITypeDescriptorContext context,
6	IServiceProvider provider, object value);
7	[C++] public: Object* EditValue(ITypeDescriptorContext* context,
8	IServiceProvider* provider, Object* value);
9	[VB] Overrides Public Function EditValue(ByVal context As
10	ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As
11	Object) As Object
12	[JScript] public override function EditValue(context: ITypeDescriptorContext,
13	provider : IServiceProvider, value : Object) : Object;
14	
15	Description
16	Edits the value of the specified data binding collection using the specified
17	service provider and context.
18	Return Value: The new collection. An
19	System.ComponentModel.ITypeDescriptorContext that identifies the
20	component or control the collection belongs to. The System.IServiceProvider to
21	use. The collection to edit.
22	GetEditStyle
23	
24	[C#] public override UITypeEditorEditStyle
25	GetEditStyle(ITypeDescriptorContext context);

1	[C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext*
2	context);
3	[VB] Overrides Public Function GetEditStyle(ByVal context As
4	ITypeDescriptorContext) As UITypeEditorEditStyle
5	[JScript] public override function GetEditStyle(context : ITypeDescriptorContext)
6	: UITypeEditorEditStyle;
7	
8	Description
9	Gets the editor stytle used by the
10	System.Web.UI.Design.DataBindingCollectionEditor.EditValue(System.Com
11	ponentModel.ITypeDescriptorContext,System.IServiceProvider,System.Obje
12	ct) method.
13	Return Value: A System.Drawing.Design.UITypeEditorEditStyle that specifies
14	the editor edit style of the component or control. An
15	System.ComponentModel.ITypeDescriptorContext that identifies the
16	component or control to retrieve the edit style for.
17	DataBindingHandler class (System.Web.UI.Design)
18	ToString
19	
20	
21	Description
22	Provides a base class for a data binding handler.
23	DataBindingHandler
24	Example Syntax:
25	ToString

	1	
	2	[C#] protected DataBindingHandler();
	3	[C++] protected: DataBindingHandler();
	4	[VB] Protected Sub New()
	5	[JScript] protected function DataBindingHandler();
	6	DataBindControl
	7	
	8	[C#] public abstract void DataBindControl(IDesignerHost designerHost, Control
	9	control);
	10	[C++] public: virtual void DataBindControl(IDesignerHost* designerHost,
	11	Control* control) = 0;
	12	[VB] MustOverride Public Sub DataBindControl(ByVal designerHost As
	13	IDesignerHost, ByVal control As Control)
	14	[JScript] public abstract function DataBindControl(designerHost: IDesignerHost,
the fluid fluid the fluid state	15	control: Control);
	16	
	17	Description
	18	Adds this data binding to the specified control. The designer host of the
	19	document. The control to data bind to.
	20	DataBindingValueUIHandler class (System.Web.UI.Design)
	21	ToString
	22	
	23	
	24	Description
	25	Provides a UI handler for data binding values.

1	DataBindingValueUIHandler
2	Example Syntax:
3	ToString
4	
5	[C#] public DataBindingValueUIHandler();
6	[C++] public: DataBindingValueUIHandler();
7	[VB] Public Sub New()
8	[JScript] public function DataBindingValueUIHandler();
9	OnGetUIValueItem
10	
11	[C#] public void OnGetUIValueItem(ITypeDescriptorContext context,
12	PropertyDescriptor propDesc, ArrayList valueUIItemList);
13	[C++] public: void OnGetUIValueItem(ITypeDescriptorContext* context,
14	PropertyDescriptor* propDesc, ArrayList* valueUIItemList);
15	[VB] Public Sub OnGetUIValueItem(ByVal context As ITypeDescriptorContext,
16	ByVal propDesc As PropertyDescriptor, ByVal valueUIItemList As ArrayList)
17	[JScript] public function OnGetUIValueItem(context: ITypeDescriptorContext,
18	propDesc: PropertyDescriptor, valueUIItemList: ArrayList);
19	
20	Description
21	Adds a data binding for the specified property and the specified value item
22	list if the current control has data bindings and the current object does not already
23	have a binding. An System.ComponentModel.ITypeDescriptorContext object
24	that can provide additional context information. A

System.ComponentModel.PropertyDescriptor that represents the property to 1 add a data binding for. A list of items that have data bindings. 2 DataFieldConverter class (System.Web.UI.Design) 3 **ToString** 5 6 Description 7 Provides a type converter that can retrieve a list of data fields that are 8 accessible through the current component's selected data source, and convert a 9 data field name to string. 10 System.Web.UI.Design.DataFieldConverter provides methods that can 11 be used to do the following: Convert a data field name string to string. 12 DataFieldConverter 13 Example Syntax: 14 **ToString** 15 16 [C#] public DataFieldConverter(); 17 [C++] public: DataFieldConverter(); 18 [VB] Public Sub New() 19 [JScript] public function DataFieldConverter(); 20 21 Description 22 Initializes a new instance of the 23 System.Web.UI.Design.DataFieldConverter class. 24

25

CanConvertFrom

[C#] public override bool CanConvertFrom(ITypeDescriptorContext context,
Type sourceType);
[C++] public: bool CanConvertFrom(ITypeDescriptorContext* context, Type*
sourceType);
[VB] Overrides Public Function CanConvertFrom(ByVal context As
ITypeDescriptorContext, ByVal sourceType As Type) As Boolean
[JScript] public override function CanConvertFrom(context:
ITypeDescriptorContext, sourceType: Type): Boolean;

Description

Gets a value indicating whether the converter can convert an object of the specified source type to the native type of the converter.

Return Value: true if the converter can perform the conversion; otherwise, false.

This method returns **true** if the source type is a string. Otherwise, this method always returns **false**. An

**System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A **System.Type** that represents the type you wish to convert from.

ConvertFrom

[C#] public override object ConvertFrom(ITypeDescriptorContext context,
 CultureInfo culture, object value);
 [C++] public: Object\* ConvertFrom(ITypeDescriptorContext\* context,
 CultureInfo\* culture, Object\* value);

1	[VB] Overrides Public Function ConvertFrom(ByVal context As
2	ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
3	As Object
4	[JScript] public override function ConvertFrom(context : ITypeDescriptorContext
5	culture : CultureInfo, value : Object) : Object;
6	
7	Description
8	Converts the specified object to the native type of the converter.
9	Return Value: An System.Object that represents the specified object after
10	conversion.
11	Override this method to provide your own conversion requirements. An
12	System.ComponentModel.ITypeDescriptorContext that can be used to gain
13	additional context information. A System.Globalization.CultureInfo that can be
14	used to support localization features. The System.Object to convert.
15	GetStandardValues
16	
17	[C#] public override StandardValuesCollection
18	GetStandardValues(ITypeDescriptorContext context);
19	[C++] public: StandardValuesCollection*
20	GetStandardValues(ITypeDescriptorContext* context);
21	[VB] Overrides Public Function GetStandardValues(ByVal context As
22	ITypeDescriptorContext) As StandardValuesCollection
23	[JScript] public override function GetStandardValues(context:
24	ITypeDescriptorContext): StandardValuesCollection;

Description

3

5

7

8

9

10

11

12

13

15

17

18

19

20

21

22

23

24

Gets the data fields present within the selected data source if information about them is available.

Return Value: A

System.ComponentModel.TypeConverter.StandardValuesCollection listing the standard accessible data sources. An

System.ComponentModel.ITypeDescriptorContext indicating the component or control to get values for.

**GetStandardValuesExclusive** 

[C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext context);

[C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext\*
context);

[VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As ITypeDescriptorContext) As Boolean

[JScript] public override function GetStandardValuesExclusive(context:

ITypeDescriptorContext): Boolean;

Description

Gets a value indicating whether the collection of standard values returned from System.ComponentModel.TypeConverter.GetStandardValues is an exclusive list of all possible values.

Return Value: true if the

System.ComponentModel.TypeConverter.StandardValuesCollection returned from System.ComponentModel.TypeConverter.GetStandardValues is an all exclusive list of all possible values; false if other values are possible.

If the list is exclusive, such as in an enumeration data type, then no other values are valid. If the list is not exclusive, then there are other valid values besides the list of standard values that

System.ComponentModel.TypeConverter.GetStandardValues provides. An System.ComponentModel.ITypeDescriptorContext that can be used to gain additional context information.

GetStandardValuesSupported

[C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext context);

[C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetStandardValuesSupported(ByVal context As ITypeDescriptorContext) As Boolean

 $[JScript]\ public\ override\ function\ GetStandard Values Supported (context:$ 

ITypeDescriptorContext): Boolean; Gets a value indicating whether the converter supports a standard set of values that can be picked from a list.

## Description

Gets a value indicating whether the converter supports a standard set of values that can be picked from a list.

Return Value: true if

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Description

System.ComponentModel.TypeConverter.GetStandardValues can be called to find a common set of values the object supports; otherwise, false. An System.ComponentModel.ITypeDescriptorContext that can be used to gain additional context information. DataMemberConverter class (System.Web.UI.Design) **ToString** Description Provides a type converter that can retrieve a list of data members that are accessible through the current component's selected data source, and convert a data member name to string. System.Web.UI.Design.DataMemberConverter provides methods that can do the following: Convert a data member name string to string. DataMemberConverter Example Syntax: **ToString** [C#] public DataMemberConverter(); [C++] public: DataMemberConverter(); [VB] Public Sub New() [JScript] public function DataMemberConverter();

2

3

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Initializes a new instance of the

# System.Web.UI.Design.DataFieldConverter class.

CanConvertFrom

[C#] public override bool CanConvertFrom(ITypeDescriptorContext context, Type sourceType);

[C++] public: bool CanConvertFrom(ITypeDescriptorContext\* context, Type\* sourceType);

[VB] Overrides Public Function CanConvertFrom(ByVal context As ITypeDescriptorContext, ByVal sourceType As Type) As Boolean [JScript] public override function CanConvertFrom(context:

ITypeDescriptorContext, sourceType: Type): Boolean;

## Description

Gets a value indicating whether the converter can convert an object of the specified source type to the native type of the converter.

Return Value: true if the converter can perform the conversion; otherwise, false.

Override this method to provide your own conversion requirements. An System.ComponentModel.ITypeDescriptorContext that can be used to gain additional context information. A System.Type that represents the type you wish to convert from.

## ConvertFrom

[C#] public override object ConvertFrom(ITypeDescriptorContext context, CultureInfo culture, object value);

1	[C++] public: Object* ConvertFrom(ITypeDescriptorContext* context,
2	CultureInfo* culture, Object* value);
3	[VB] Overrides Public Function ConvertFrom(ByVal context As
4	ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
5	As Object
6	[JScript] public override function ConvertFrom(context : ITypeDescriptorContext,
7	culture : CultureInfo, value : Object) : Object;
8	
9	Description
10	Converts the specified object to the native type of the converter.
11	Return Value: An System.Object that represents the specified object after
12	conversion.
13	Override this method to provide your own conversion requirements. An
14	System.ComponentModel.ITypeDescriptorContext that can be used to gain
15	additional context information. A System.Globalization.CultureInfo that can be
16	used to support localization features. The System.Object to convert.
17	GetStandardValues
18	
19	[C#] public override StandardValuesCollection
20	GetStandardValues(ITypeDescriptorContext context);
21	[C++] public: StandardValuesCollection*
22	GetStandardValues(ITypeDescriptorContext* context);
23	[VB] Overrides Public Function GetStandardValues(ByVal context As
24	ITypeDescriptorContext) As StandardValuesCollection
25	[JScript] public override function GetStandardValues(context:

11
1
r ,alle
i selle

	1	ITypeDescriptorContext): StandardValuesCollection;
	2	
	3	Description
	4	Gets the data members present within the selected data source, if
	5	information about them is available.
	6	Return Value: A
	7	System.ComponentModel.TypeConverter.StandardValuesCollection listing
	8	the standard accessible data sources. An
	9	System.ComponentModel.ITypeDescriptorContext indicating the component
of only	10	or control to get values for.
	11	GetStandardValuesExclusive
	12	
ms.	13	[C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext
	14	context);
The state of the s	15	[C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext*
1 1007 1 1007 1 1007 1 1007	16	context);
•	17	[VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As
	18	ITypeDescriptorContext) As Boolean
	19	[JScript] public override function GetStandardValuesExclusive(context:
	20	ITypeDescriptorContext): Boolean;
	21	
	22	Description
	23	Gets a value indicating whether the collection of standard values returned
	24	from System.ComponentModel.TypeConverter.GetStandardValues is an
	25	exclusive list of all possible values.

Return Value: true if the

System.ComponentModel.TypeConverter.StandardValuesCollection returned from System.ComponentModel.TypeConverter.GetStandardValues is an exclusive list of possible values; false if other values are possible.

If the list is exclusive, such as in an enumeration data type, then no other values are valid. If the list is not exclusive, then there are other valid values besides the list of standard values that

System.ComponentModel.TypeConverter.GetStandardValues provides. An System.ComponentModel.ITypeDescriptorContext that provides a format context.

GetStandardValuesSupported

[C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext context);

[C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetStandardValuesSupported(ByVal context As ITypeDescriptorContext) As Boolean

[JScript] public override function GetStandardValuesSupported(context:

ITypeDescriptorContext): Boolean; Gets a value indicating whether the converter supports a standard set of values that can be picked from a list.

Description

Gets a value indicating whether the converter supports a standard set of values that can be picked from a list.

1	Return Value: true if
2	System.ComponentModel.TypeConverter.GetStandardValues should be
3	called to find a common set of values the object supports; otherwise, false. An
4	System.ComponentModel.ITypeDescriptorContext that can be used to gain
5	additional context information.
6	DataSourceConverter class (System.Web.UI.Design)
7	ToString
8	
9	
10	Description
11	Provides a type converter that can retrieve a list of data sources accessible
12	to the current component, and convert a data source name to string.
13	System.Web.UI.Design.DataSourceConverter provides methods that can
14	be used to do the following: Convert a data source name string to string.
15	DataSourceConverter
16	Example Syntax:
17	ToString
18	
19	[C#] public DataSourceConverter();
20	[C++] public: DataSourceConverter();
21	[VB] Public Sub New()
22	[JScript] public function DataSourceConverter();
23	
24	Description
25	

The first that the

1

2

3

5

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Initializes a new instance of the

System.Web.UI.Design.DataSourceConverter class.

CanConvertFrom

[C#] public override bool CanConvertFrom(ITypeDescriptorContext context, Type sourceType);

[C++] public: bool CanConvertFrom(ITypeDescriptorContext\* context, Type\* sourceType);

[VB] Overrides Public Function CanConvertFrom(ByVal context As ITypeDescriptorContext, ByVal sourceType As Type) As Boolean [JScript] public override function CanConvertFrom(context:

ITypeDescriptorContext, sourceType: Type): Boolean;

Description

Gets a value indicating whether the converter can convert an object of the specified source type to the native type of the converter.

Return Value: true if the converter can perform the conversion; otherwise, false.

Override this method to provide your own conversion requirements. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A **System.Type** that represents the type you wish to convert from.

ConvertFrom

[C#] public override object ConvertFrom(ITypeDescriptorContext context, CultureInfo culture, object value);

MS1-863US.APP

17

18

19

20

21

22

23

24

i	[C++] public: Object* ConvertFrom(ITypeDescriptorContext* context,
2	CultureInfo* culture, Object* value);
3	[VB] Overrides Public Function ConvertFrom(ByVal context As
4	ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
5	As Object
6	[JScript] public override function ConvertFrom(context : ITypeDescriptorContext,
7	culture : CultureInfo, value : Object) : Object;
8	
9	Description
10	Converts the specified object to the native type of the converter.
11	Return Value: An System.Object that represents the specified object after
12	conversion.
13	Override this method to provide your own conversion requirements. An
14	System.ComponentModel.ITypeDescriptorContext that can be used to gain
15	additional context information. The System.Globalization.CultureInfo that can

GetStandardValues

[C#] public override StandardValuesCollection

GetStandardValues(ITypeDescriptorContext context);

[C++] public: StandardValuesCollection\*

GetStandardValues(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetStandardValues(ByVal context As

be used to support localization features. The System.Object to convert.

ITypeDescriptorContext) As StandardValuesCollection

 $[JScript]\ public\ override\ function\ GetStandardValues (context:$ 

1	ITypeDescriptorContext): StandardValuesCollection;
2	
3	Description
4	Gets the standard data sources accessible to the control.
5	Return Value: A
6	System.ComponentModel.TypeConverter.StandardValuesCollection listing
7	the standard accessible data sources. An
8	System.ComponentModel.ITypeDescriptorContext indicating the component
9	or control to get values for.
10	GetStandardValuesExclusive
11	
12	[C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext
13	context);
14	[C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext*
15	context);
16	[VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As
17	ITypeDescriptorContext) As Boolean
18	[JScript] public override function GetStandardValuesExclusive(context:
19	ITypeDescriptorContext): Boolean;
20	
21	Description
22	Gets a value indicating whether the collection of standard values returned
23	from System.ComponentModel.TypeConverter.GetStandardValues is an
24	exclusive list of all possible values.
25	Return Value: true if the

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

System.ComponentModel.TypeConverter.StandardValuesCollection returned from System.ComponentModel.TypeConverter.GetStandardValues is an exclusive list of all possible values; false if other values are possible.

If the list is exclusive, such as in an enumeration data type, then no other values are valid. If the list is not exclusive, then there are other valid values besides the list of standard values that

System.ComponentModel.TypeConverter.GetStandardValues provides. An System.ComponentModel.ITypeDescriptorContext that can be used to gain additional context information.

GetStandardValuesSupported

[C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext context);

[C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetStandardValuesSupported(ByVal context As ITypeDescriptorContext) As Boolean

 $[JScript]\ public\ override\ function\ GetStandardValuesSupported (context:$ 

ITypeDescriptorContext): Boolean; Gets a value indicating whether the converter supports a standard set of values that can be picked from a list.

Description

Gets a value indicating whether the converter supports a standard set of values that can be picked from a list.

Return Value: true if

Description

System.ComponentModel.TypeConverter.GetStandardValues should be		
called to find a common set of values the object supports; otherwise, false. An		
System.ComponentModel.ITypeDescriptorContext that can be used to gain		
additional context information.		
DesignTimeData class (System.Web.UI.Design)		
ToString		
Description		
Provides helper methods that can be used by control designers to generate		
sample data for use in design time databinding.		
ToString		
[C#] public static readonly EventHandler DataBindingHandler;		
[C++] public: static EventHandler* DataBindingHandler;		
[VB] Public Shared ReadOnly DataBindingHandler As EventHandler		
[JScript] public static var DataBindingHandler: EventHandler;		
CreateDummyDataTable		
[C#] public static DataTable CreateDummyDataTable();		
[C++] public: static DataTable* CreateDummyDataTable();		
[VB] Public Shared Function CreateDummyDataTable() As DataTable		
[IScript] public static function CreateDummyDataTable() : DataTable:		

Creates a datatable that contains sample data.

Return Value: A new System.Data.DataTable that contains 3 columns of type string.

CreateSampleDataTable

[C#] public static DataTable CreateSampleDataTable(IEnumerable referenceData);

[C++] public: static DataTable\* CreateSampleDataTable(IEnumerable\* referenceData);

[VB] Public Shared Function CreateSampleDataTable(ByVal referenceData As IEnumerable) As DataTable

[JScript] public static function CreateSampleDataTable(referenceData:

IEnumerable): DataTable;

Description

Creates a sample datatable with the same schema as the supplied datasource.

Return Value: A sample data table that contains dummy data in the format of the specified data source, or a default dummy data table if the specified data source contained no data columns. A data source with the schema to use as the format for the sample datatable.

GetDataFields

[C#] public static PropertyDescriptorCollection GetDataFields(IEnumerable dataSource);

1	[C++] public: static PropertyDescriptorCollection* GetDataFields(IEnumerable*
2	dataSource);
3	[VB] Public Shared Function GetDataFields(ByVal dataSource As IEnumerable)
4	As PropertyDescriptorCollection
5	[JScript] public static function GetDataFields(dataSource : IEnumerable) :
6	PropertyDescriptorCollection;
7	
8	Description
9	Gets the data fields of the specified data source.
10	Return Value: A System.ComponentModel.PropertyDescriptorCollection that
11	represents the data fields of the speciifed data source. The data source to retrieve
12	the data fields of.
13	GetDataMember
14	
15	[C#] public static IEnumerable GetDataMember(IListSource dataSource, string
16	dataMember);
17	[C++] public: static IEnumerable* GetDataMember(IListSource* dataSource,
18	String* dataMember);
19	[VB] Public Shared Function GetDataMember(ByVal dataSource As IListSource
20	ByVal dataMember As String) As IEnumerable
21	[JScript] public static function GetDataMember(dataSource : IListSource,
22	dataMember : String) : IEnumerable;
23	
24	Description
25	

MS1-863US.APP

Gets the specified data member from the specified data source.

Return Value: The specified data member from the specified data source, if it exists.

This method searches the specified data source for the specified data member. If the *dataMember* property is **null**, the first item in the specified data source is returned. The data source that contains the member to retrieve. The data member to retrieve.

#### GetDataMembers

[C#] public static string[] GetDataMembers(object dataSource);
[C++] public: static String* GetDataMembers(Object* dataSource)gc[];
[VB] Public Shared Function GetDataMembers(ByVal dataSource As Object) As
String()

[JScript] public static function GetDataMembers(dataSource : Object) : String[];

# Description

Gets the data members of the specified data source.

Return Value: A string array that represents the data members of the specified data source. The data source to retrieve the members of.

## GetDesignTimeDataSource

[C#] public static IEnumerable GetDesignTimeDataSource(DataTable dataTable,
int minimumRows);
[C++] public: static IEnumerable\* GetDesignTimeDataSource(DataTable\*
dataTable, int minimumRows);

[VB] Public Shared Function GetDesignTimeDataSource(ByVal dataTable As DataTable, ByVal minimumRows As Integer) As IEnumerable 2 [JScript] public static function GetDesignTimeDataSource(dataTable: DataTable, 3 minimumRows: int): IEnumerable; 5 Description 6 Adds sample rows to the specified data table and returns the updated table. 7 Return Value: A live data source for use at design time. The table that defines the 8 shape of the data source. Sample rows are added to this table. The minimum 9 number of rows of sample data that the data source will contain. 10 **GetSelectedDataSource** 11 12 [C#] public static object GetSelectedDataSource(IComponent component, string 13 dataSource); 14 [C++] public: static Object\* GetSelectedDataSource(IComponent\* component, 15 String\* dataSource); [VB] Public Shared Function GetSelectedDataSource(ByVal component As 17 IComponent, ByVal dataSource As String) As Object 18 [JScript] public static function GetSelectedDataSource(component: IComponent, 19 dataSource: String): Object; Gets the specified data source. 20 21 Description 22

23

24

25

Gets the specified data source.

Return Value: The data source object, or **null** if the data source, specified

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

component's site, or the container of the data source could not be accessed. The component that contains the data source. The data source to retrieve. GetSelectedDataSource [C#] public static IEnumerable GetSelectedDataSource(IComponent component, string dataSource, string dataMember); [C++] public: static IEnumerable\* GetSelectedDataSource(IComponent\* component, String\* dataSource, String\* dataMember); [VB] Public Shared Function GetSelectedDataSource(ByVal component As IComponent, ByVal dataSource As String, ByVal dataMember As String) As **IEnumerable** [JScript] public static function GetSelectedDataSource(component : IComponent, dataSource: String, dataMember: String): IEnumerable; Description Gets the specified data member of the specified data source. Return Value: The data member, or **null** if the data source could not be accessed. The component that contains the data source. The data source to retrieve. The data member to retrieve. HtmlControlDesigner class (System.Web.UI.Design) **ToString** 

Description

Provides basic design-time functionality for ASP.NET server controls.

```
HtmlControlDesigner
           Example Syntax:
           ToString
    [C#] public HtmlControlDesigner();
    [C++] public: HtmlControlDesigner();
    [VB] Public Sub New()
 7
    [JScript] public function HtmlControlDesigner();
 8
9
    Description
10
           Initializes a new instance of the
11
    System. Web. UI. Design. Html Control Designer class.
12
           AssociatedComponents
13
           Behavior
14
           ToString
15
16
17
    Description
18
           Gets or sets the DHTML behavior associated with the designer instance.
19
           Component
20
           DataBindings
21
           ToString
22
23
24
    Description
```

	1	Gets or sets the data bindings for the current control.
	2	DesignTimeElement
	3	ToString
	4	
	5	[C#] protected object DesignTimeElement {get;}
	6	[C++] protected:property Object* get_DesignTimeElement();
	7	[VB] Protected ReadOnly Property DesignTimeElement As Object
	8	[JScript] protected function get DesignTimeElement() : Object;
	9	
1991	10	Description
આ મામ માત્રા પૈકાલ પૈકાર્થ પૈકાર્થ પૈકાર્થ પૈકાર્થ	11	Gets the base Web Forms designer associated to the ASPX document.
oly man de	12	InheritanceAttribute
177 July 1110	13	Inherited
	14	ShadowProperties
	15	ShouldCodeSerialize
	16	ToString
	17	
	18	
	19	Description
	20	Indicates whether a field declaration for the control should be created in the
	21	code-behind file for the current design document.
	22	Verbs
	23	Dispose
	24	
	25	[C#] protected override void Dispose(bool disposing);

1	[C++] protected: void Dispose(bool disposing);
2	[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)
3	[JScript] protected override function Dispose(disposing : Boolean);
4	
5	Description
6	Releases the unmanaged resources used by the
7	System.Web.UI.Design.HtmlControlDesigner and optionally releases the
8	managed resources.
9	This method is called by the public <b>Dispose()</b> method and the
10	System.Object.Finalize method. true to release both managed and unmanaged
11	resources; false to release only unmanaged resources.
12	
13	Description
14	Initializes the designer and sets the component for design.
15	System.Web.UI.Design.ControlDesigner.Initialize(System.Component
16	Model.IComponent) should be called by the designer host to initialize the
17	designer. The control element for design.
18	OnBehaviorAttached
19	
20	[C#] protected virtual void OnBehaviorAttached();
21	[C++] protected: virtual void OnBehaviorAttached();
22	[VB] Overridable Protected Sub OnBehaviorAttached()
23	[JScript] protected function OnBehaviorAttached();
24	
25	Description

Notification that is called when a behavior is attached to the designer.

OnBehaviorDetaching

[C#] protected virtual void OnBehaviorDetaching();

[C++] protected: virtual void OnBehaviorDetaching();

[JScript] protected function OnBehaviorDetaching();

[VB] Overridable Protected Sub OnBehaviorDetaching()

## Description

Notification that is called when a behavior is detatched from the designer.

OnBindingsCollectionChanged

[C#] protected virtual void OnBindingsCollectionChanged(string propName);[C++] protected: virtual void OnBindingsCollectionChanged(String\* propName);[VB] Overridable Protected Sub OnBindingsCollectionChanged(ByVal propName As String)

[JScript] protected function OnBindingsCollectionChanged(propName : String);

## Description

Provides a method that can be used to indicate when a data binding has been changed.

This method should be called by a control designer after it has made a change to the data bindings for the control. The name of the property that has been changed.

**OnSetParent** 

[C#] public virtual void OnSetParent();
[C++] public: virtual void OnSetParent();
[VB] Overridable Public Sub OnSetParent()
[JScript] public function OnSetParent();

Description

Notification that is called when the associated control is parented.

**PreFilterEvents** 

[C#] protected override void PreFilterEvents(IDictionary events);

[C++] protected: void PreFilterEvents(IDictionary\* events);

[VB] Overrides Protected Sub PreFilterEvents(ByVal events As IDictionary)

[JScript] protected override function PreFilterEvents(events : IDictionary);

# Description

Allows a designer to filter the set of member events that the component it is designing will expose through a **System.ComponentModel.TypeDescriptor** object.

If you are overriding this method you should call the base implementation before you perform your own filtering. When overriding this method, it should return the augmented set of attributes. If the method does not modify any attributes, it may just return a reference to its input parameter. If you do make a change to the attributes, you must create a new array. The set of events for the component.

# **PreFilterProperties**

[C#] protected override void PreFilterProperties(IDictionary properties);

[C++] protected: void PreFilterProperties(IDictionary\* properties);

[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As IDictionary)

[JScript] protected override function PreFilterProperties(properties: IDictionary);

## Description

Allows a designer to filter the set of member attributes that the component it is designing will expose through a **System.ComponentModel.TypeDescriptor** object.

If you are overriding this method you should call the base implementation before you perform your own filtering. When overriding this method, it should return the augmented set of attributes. If the method does not modify any attributes, it may just return a reference to its input parameter. If you do make a change to the attributes, you must create a new array. The set of properties to filter for the component.

HtmlIntrinsicControlDesigner class (System.Web.UI.Design)
ToString

## Description

Provides a base designer for all intrinsic HTML controls.

1	Intrinsic HTML controls are defined within, or derived from classes
2	defined within, the System. Web. UI. Html Controls. Html Control namespace.
3	HtmlIntrinsicControlDesigner
4	Example Syntax:
5	ToString
6	
7	[C#] public HtmlIntrinsicControlDesigner();
8	[C++] public: HtmlIntrinsicControlDesigner();
9	[VB] Public Sub New()
10	[JScript] public function HtmlIntrinsicControlDesigner();
11	AssociatedComponents
12	Behavior
13	Component
14	DataBindings
15	DesignTimeElement
16	InheritanceAttribute
17	Inherited
18	ShadowProperties
19	ShouldCodeSerialize
20	Verbs
21	HyperLinkDataBindingHandler class (System.Web.UI.Design)
22	ToString
23	
24	
25	Description

Provides a data binding handler for a hyperlink property. HyperLinkDataBindingHandler 2 Example Syntax: **ToString** 5 [C#] public HyperLinkDataBindingHandler(); 6 [C++] public: HyperLinkDataBindingHandler(); [VB] Public Sub New() 8 [JScript] public function HyperLinkDataBindingHandler(); 9 DataBindControl 10 11 [C#] public override void DataBindControl(IDesignerHost designerHost, Control 12 control); 13 [C++] public: void DataBindControl(IDesignerHost\* designerHost, Control\* 14 control); 15 [VB] Overrides Public Sub DataBindControl(ByVal designerHost As 16 IDesignerHost, ByVal control As Control) 17 [JScript] public override function DataBindControl(designerHost: IDesignerHost, 18 control: Control); 19 20 Description 21 Adds this data binding to the specified control. The designer host for the 22 document that contains the control. The control to add this data binding to. 23 IControlDesignerBehavior interface (System.Web.UI.Design) 24 **ToString** 25

1	
2	
3	Description
4	Provides an interface to provide a behavior for a control designer.
5	DesignTimeElementView
6	ToString
7	
8	[C#] object DesignTimeElementView {get;}
9	[C++] Object* get_DesignTimeElementView();
10	[VB] ReadOnly Property DesignTimeElementView As Object
11	[JScript] abstract function get DesignTimeElementView() : Object;
12	
13	Description
14	Gets or sets the design-time view control object for the designer.
15	DesignTimeHtml
16	ToString
17	
18	[C#] string DesignTimeHtml {get; set;}
19	[C++] String* get_DesignTimeHtml();void set_DesignTimeHtml(String*);
20	[VB] Property DesignTimeHtml As String
21	[JScript] abstract function get DesignTimeHtml(): String;public abstract function
22	set DesignTimeHtml(String);
23	
24	Description
25	Gets or sets the design-time HTML for the designer's control.

1	OnTemplateModeChanged
2	
3	[C#] void OnTemplateModeChanged();
4	[C++] void OnTemplateModeChanged();
5	[VB] Sub OnTemplateModeChanged()
6	[JScript] function OnTemplateModeChanged();
7	
8	Description
9	Raises the TemplateModeChanged event.
10	Raising an event invokes the event handler through a delegate. For more
11	information, see .
12	IDataSourceProvider interface (System.Web.UI.Design)
13	OnTemplateModeChanged
14	
15	
16	Description
17	Provides an interface that enables access to a data source.
18	A class that can provide a data source can implement this interface to
19	enable access to its data source by objects that use the
20	System.Web.UI.Design.IDataSourceProvider interface.
21	GetResolvedSelectedDataSource
22	
23	[C#] IEnumerable GetResolvedSelectedDataSource();
24	[C++] IEnumerable* GetResolvedSelectedDataSource();
25	[VB] Function GetResolvedSelectedDataSource() As IEnumerable

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Designer

[JScript] function GetResolvedSelectedDataSource(): IEnumerable; Description When implemented in a derived class, gets the selected data member from the selected data source. Return Value: The selected data member from the selected data source. GetSelectedDataSource [C#] object GetSelectedDataSource(); [C++] Object\* GetSelectedDataSource(); [VB] Function GetSelectedDataSource() As Object [JScript] function GetSelectedDataSource(): Object; Description When implemented in a derived class, gets a reference to the data source that this data source provider provides. *Return Value:* The data source that this data source provider provides. IHtmlControlDesignerBehavior interface (System.Web.UI.Design) GetSelectedDataSource Description Provides an interface to provide a behavior for an Interactive HTML control designer.

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

GetSelectedDataSource

[C#] HtmlControlDesigner Designer {get; set;}

[C++] HtmlControlDesigner\* get\_Designer();void

set Designer(HtmlControlDesigner\*);

[VB] Property Designer As HtmlControlDesigner

[JScript] abstract function get Designer(): HtmlControlDesigner; public abstract

function set Designer(HtmlControlDesigner);

# Description

Gets or sets the designer that this behavior is associated with.

DesignTimeElement

GetSelectedDataSource

[C#] object DesignTimeElement {get;}

[C++] Object\* get DesignTimeElement();

[VB] ReadOnly Property DesignTimeElement As Object

[JScript] abstract function get DesignTimeElement(): Object;

# Description

Gets the element that this designer is designing.

GetAttribute

[C#] object GetAttribute(string attribute, bool ignoreCase);

 $[C++]\ Object *\ Get Attribute (String *\ attribute,\ bool\ ignore Case);$ 

1	[VB] Function GetAttribute(ByVal attribute As String, ByVal ignoreCase As
2	Boolean) As Object
3	[JScript] function GetAttribute(attribute: String, ignoreCase: Boolean): Object;
4	
5	Description
6	Gets the specified attribute.
7	Return Value: The attribute that was retrieved. The attribute to retrieve. true if the
8	attribute syntax is case-insensitive; otherwise, false.
9	GetStyleAttribute
10	
11	[C#] object GetStyleAttribute(string attribute, bool designTimeOnly, bool
12	ignoreCase);
13	[C++] Object* GetStyleAttribute(String* attribute, bool designTimeOnly, bool
14	ignoreCase);
15	[VB] Function GetStyleAttribute(ByVal attribute As String, ByVal
16	designTimeOnly As Boolean, ByVal ignoreCase As Boolean) As Object
17	[JScript] function GetStyleAttribute(attribute: String, designTimeOnly: Boolean,
18	ignoreCase : Boolean) : Object;
19	
20	Description
21	Gets the specified style attribute.
22	Return Value: The style attribute that was retrieved. The style attribute to retrieve.
23	true if the attribute is a design-time only attribute; otherwise, false. true if the
24	attribute syntax is case-insensitive; otherwise, false.
25	RemoveAttribute

[C#] void RemoveAttribute(string attribute, bool ignoreCase); 2 [C++] void RemoveAttribute(String\* attribute, bool ignoreCase); 3 [VB] Sub RemoveAttribute(ByVal attribute As String, ByVal ignoreCase As 4 Boolean) 5 [JScript] function RemoveAttribute(attribute: String, ignoreCase: Boolean); 6 7 Description 8 Removes the specified attribute. The attribute to remove. true if the 9 attribute syntax is case-insensitive; otherwise, false. 10 RemoveStyleAttribute 11 12 [C#] void RemoveStyleAttribute(string attribute, bool designTimeOnly, bool 13 ignoreCase); 14 [C++] void RemoveStyleAttribute(String\* attribute, bool designTimeOnly, bool 15 ignoreCase); 16 [VB] Sub RemoveStyleAttribute(ByVal attribute As String, ByVal 17 designTimeOnly As Boolean, ByVal ignoreCase As Boolean) 18 [JScript] function RemoveStyleAttribute(attribute: String, designTimeOnly: 19 Boolean, ignoreCase: Boolean); 20 21 Description 22 Removes the specified style attribute. The style attribute to remove. true if 23 the attribute is a design-time only attribute; otherwise, false. true if the attribute 24

syntax is case-insensitive; otherwise, false.

## SetAttribute

[C#] void SetAttribute(string attribute, object value, bool ignoreCase);

[C++] void SetAttribute(String\* attribute, Object\* value, bool ignoreCase);

[VB] Sub SetAttribute(ByVal attribute As String, ByVal value As Object, ByVal ignoreCase As Boolean)

[JScript] function SetAttribute(attribute : String, value : Object, ignoreCase : Boolean);

## Description

Sets the specified attribute to the specified object. The attribute to set. The object on which to set the attribute. **true** if the attribute syntax is case-insensitive; otherwise, **false**.

# SetStyleAttribute

[C#] void SetStyleAttribute(string attribute, bool designTimeOnly, object value, bool ignoreCase);

[C++] void SetStyleAttribute(String\* attribute, bool designTimeOnly, Object\* value, bool ignoreCase);

[VB] Sub SetStyleAttribute(ByVal attribute As String, ByVal designTimeOnly As Boolean, ByVal value As Object, ByVal ignoreCase As Boolean)

[JScript] function SetStyleAttribute(attribute : String, designTimeOnly : Boolean, value : Object, ignoreCase : Boolean);

MS1-863US.APP

Description

Sets the specified style attribute to the specified object. The attribute to set.
true if the attribute is a design-time only attribute; otherwise, false. The object to
set the attribute on. true if the attribute syntax is case-insensitive; otherwise, false.
ImageUrlEditor class (System.Web.UI.Design)
SetStyleAttribute
Description
Provides a user interface for selecting a URL.
This class extends System.Web.UI.Design.UrlEditor and provides a
caption and filter property.
ImageUrlEditor
Example Syntax:
SetStyleAttribute
[C#] public ImageUrlEditor();
[C++] public: ImageUrlEditor();
[VB] Public Sub New()
[JScript] public function ImageUrlEditor();
Caption
SetStyleAttribute
[C#] protected override string Caption {get;}
[C++] protected:property virtual String* get_Caption();

[VB] Overrides Protected ReadOnly Property Caption As String

1	[JScript] protected function get Caption() : String;
2	
3	Description
4	Gets the caption for the editor.
5	Filter
6	SetStyleAttribute
7	
8	[C#] protected override string Filter {get;}
9	[C++] protected:property virtual String* get_Filter();
10	[VB] Overrides Protected ReadOnly Property Filter As String
11	[JScript] protected function get Filter(): String;
12	
13	Description
14	Gets the filter to use for filtering the file list.
15	Options
16	ITemplateEditingFrame interface (System.Web.UI.Design)
17	ToString
18	
19	
20	Description
21	Provides an interface to manage a template editing area.
22	ControlStyle
23	ToString
24	
25	[C#] Style ControlStyle {get;}

```
[C++] Style* get_ControlStyle();
    [VB] ReadOnly Property ControlStyle As Style
2
    [JScript] abstract function get ControlStyle(): Style;
3
4
    Description
5
           Gets or sets the style for the editing frame.
6
           InitialHeight
7
           ToString
8
9
    [C#] int InitialHeight {get; set;}
10
    [C++] int get InitialHeight(); void set_InitialHeight(int);
11
    [VB] Property InitialHeight As Integer
12
    [JScript] abstract function get InitialHeight(): int;public abstract function set
13
    InitialHeight(int);
14
15
    Description
16
            Gets or sets the initial height for the control.
17
            InitialWidth
18
            ToString
19
20
    [C#] int InitialWidth {get; set;}
21
    [C++] int get InitialWidth(); void set InitialWidth(int);
22
     [VB] Property InitialWidth As Integer
23
    [JScript] abstract function get InitialWidth(): int;public abstract function set
24
    InitialWidth(int);
```

1	
2	Description
3	Gets or sets the initial width for the control.
4	Name
5	ToString
6	
7	[C#] string Name {get;}
8	[C++] String* get_Name();
9	[VB] ReadOnly Property Name As String
10	[JScript] abstract function get Name(): String;
11	
12	Description
13	Gets or sets the name for this editing frame.
14	TemplateNames
15	ToString
16	
17	[C#] string[] TemplateNames {get;}
18	[C++] String* get_TemplateNames();
19	[VB] ReadOnly Property TemplateNames As String ()
20	[JScript] abstract function get TemplateNames(): String[];
21	
22	Description
23	Gets or sets a set of names of templates to use.
24	TemplateStyles
25	ToString

```
[C#] Style[] TemplateStyles {get;}
    [C++] Style* get TemplateStyles();
    [VB] ReadOnly Property TemplateStyles As Style ()
    [JScript] abstract function get TemplateStyles(): Style[];
5
6
    Description
7
           Gets or sets the template styles for this control.
8
           Verb
9
           ToString
10
11
    [C#] TemplateEditingVerb Verb {get; set;}
12
    [C++] TemplateEditingVerb* get_Verb();void set_Verb(TemplateEditingVerb*);
13
    [VB] Property Verb As TemplateEditingVerb
14
    [JScript] abstract function get Verb(): TemplateEditingVerb;public abstract
15
    function set Verb(TemplateEditingVerb);
16
17
    Description
18
           Gets or sets the verb to invoke when editing the template.
19
           Close
20
21
    [C#] void Close(bool saveChanges);
22
    [C++] void Close(bool saveChanges);
23
    [VB] Sub Close(ByVal saveChanges As Boolean)
24
     [JScript] function Close(saveChanges : Boolean);
25
```

Description

3

5

6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Closes the control and saves any changes to the control if the specified value is **true** . **true** if any changes to the document should be changed; otherwise, **false**.

Open

[C#] void Open();

[C++] void Open();

[VB] Sub Open()

[JScript] function Open();

Description

Opens and displays the control.

Resize

[C#] void Resize(int width, int height);

[C++] void Resize(int width, int height);

[VB] Sub Resize(ByVal width As Integer, ByVal height As Integer)

[JScript] function Resize(width: int, height: int);

Description

Resizes the control to the specified width and height. The new width for the control. The new height for the control.

Save

lee@hayes plic 509+324+9256

1	FO//7 :10 O
2	[C#] void Save();
3	[C++] void Save();
4	[VB] Sub Save()
5	[JScript] function Save();
6	
7	Description
8	Saves any changes to the document.
9	UpdateControlName
10	
11	[C#] void UpdateControlName(string newName);
12	[C++] void UpdateControlName(String* newName);
13	[VB] Sub UpdateControlName(ByVal newName As String)
14	[JScript] function UpdateControlName(newName : String);
15	
16	Description
17	Changes the name for the control to the specified name. The new name for
18	the control.
19	ITemplateEditingService interface (System.Web.UI.Design)
20	UpdateControlName
21	
22	
23	Description
24	Provides services for editing templated control templates at design-time.
25	SupportsNestedTemplateEditing
	••

## UpdateControlName

[C#] bool SupportsNestedTemplateEditing {get;}

[C++] bool get\_SupportsNestedTemplateEditing();

[VB] ReadOnly Property SupportsNestedTemplateEditing As Boolean [JScript] abstract function get SupportsNestedTemplateEditing(): Boolean;

Description

1

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Indicates whether the service supports nested template editing.

This property indicates whether a template editor can be invoked from a parent template editor for a component of the parent template editor's templated control. This is called nested template editing, because a template editor is launched for a component of a control for which a template editor is already operating.

CreateFrame

[C#] ITemplateEditingFrame CreateFrame(TemplatedControlDesigner designer, string frameName, string[] templateNames);

[C++] ITemplateEditingFrame\* CreateFrame(TemplatedControlDesigner\* designer, String\* frameName, String\* templateNames \_\_gc[]);

[VB] Function CreateFrame(ByVal designer As TemplatedControlDesigner,

ByVal frameName As String, ByVal templateNames() As String) As

ITemplateEditingFrame

[JScript] function CreateFrame(designer: TemplatedControlDesigner, frameName

: String, templateNames : String[]) : ITemplateEditingFrame; Creates a new

template editing frame.

Description

Creates a new template editing frame for the specified

System.Web.UI.Design.TemplatedControlDesigner, using the specified name and templates.

Return Value: The new System.Web.UI.Design.ITemplateEditingFrame. The designer that will use the template editing frame. The name of the editing frame that will be displayed on the frame. Typically this is the same as the System.ComponentModel.Design.DesignerVerb.Text property used as the menu text for the System.Web.UI.Design.TemplateEditingVerb that is invoked to create the frame. The names of the templates that the template editing frame will contain.

CreateFrame

[C#] ITemplateEditingFrame CreateFrame(TemplatedControlDesigner designer, string frameName, string[] templateNames, Style controlStyle, Style[] templateStyles);

[C++] ITemplateEditingFrame\* CreateFrame(TemplatedControlDesigner\* designer, String\* frameName, String\* templateNames \_\_gc[], Style\* controlStyle, Style\* templateStyles[]);

[VB] Function CreateFrame(ByVal designer As TemplatedControlDesigner, ByVal frameName As String, ByVal templateNames() As String, ByVal controlStyle As Style, ByVal templateStyles() As Style) As

ITemplateEditingFrame

[JScript] function CreateFrame(designer : TemplatedControlDesigner, frameName : String, templateNames : String[], controlStyle : Style, templateStyles : Style[]) : ITemplateEditingFrame;

### Description

Creates a new template editing frame for the specified System.Web.UI.Design.TemplatedControlDesigner, using the specified name, template names, control style, and template styles.

Return Value: The new System.Web.UI.Design.ITemplateEditingFrame. The designer that will use the template editing frame. The name of the editing frame that will be displayed on the frame. Typically this is the same as the System.ComponentModel.Design.DesignerVerb.Text property used as the menu text for the System.Web.UI.Design.TemplateEditingVerb that is invoked to create the frame. The names of the templates that the template editing frame will contain. The control style for the editing frame. The template styles for the editing frame.

GetContainingTemplateName

[C#] string GetContainingTemplateName(Control control);

[C++] String\* GetContainingTemplateName(Control\* control);

[VB] Function GetContainingTemplateName(ByVal control As Control) As String

[JScript] function GetContainingTemplateName(control: Control): String;

Gets the name of the parent template. 1 Return Value: The name of the parent template. The control to get the name of the 2 parent template of. 3 IWebFormReferenceManager interface (System.Web.UI.Design) GetContainingTemplateName 5 Description 8 Provides an interface that can be used to manage references stored by a 9 Web Form. 10 GetObjectType 11 12 [C#] Type GetObjectType(string tagPrefix, string typeName); 13 [C++] Type\* GetObjectType(String\* tagPrefix, String\* typeName); 14 [VB] Function GetObjectType(ByVal tagPrefix As String, ByVal typeName As 15 String) As Type 16 [JScript] function GetObjectType(tagPrefix : String, typeName : String) : Type; 17 18 Description 19 Gets the type of the specified object. 20 Return Value: The System. Type of the object, if it could be resolved. The tag 21 prefix for the type. The name of the type. 22 **GetRegisterDirectives** 23 24 [C#] string GetRegisterDirectives();

1	[C++] String* GetRegisterDirectives();
2	[VB] Function GetRegisterDirectives() As String
3	[JScript] function GetRegisterDirectives() : String;
4	
5	Description
6	Gets the register directives.
7	Return Value: The register directives for the project.
8	GetTagPrefix
9	
10	[C#] string GetTagPrefix(Type objectType);
11	[C++] String* GetTagPrefix(Type* objectType);
12	[VB] Function GetTagPrefix(ByVal objectType As Type) As String
13	[JScript] function GetTagPrefix(objectType : Type) : String;
14	
15	Description
16	Gets the tag prefix.
17	Return Value: The tag prefix for the specified object type, if it could be located.
18	The type of the object.
19	IWebFormsBuilderUIService interface (System.Web.UI.Design)
20	GetTagPrefix
21	
22	
23	Description
24	Provides methods to launch specific user interfaces for building properties
25	at design-time.

#### BuildColor

[C#] string BuildColor(Control owner, string initialColor);

[C++] String\* BuildColor(Control\* owner, String\* initialColor);

[VB] Function BuildColor(ByVal owner As Control, ByVal initialColor As String) As String

[JScript] function BuildColor(owner: Control, initialColor: String): String;

## Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Launches an editor to build a color property.

Return Value: A string that represents the color that was selected. The control that contains the property to build. The initial color for the editor to pre-select.

#### BuildUrl

[C#] string BuildUrl(Control owner, string initialUrl, string baseUrl, string caption, string filter, UrlBuilderOptions options);

[C++] String\* BuildUrl(Control\* owner, String\* initialUrl, String\* baseUrl,

String\* caption, String\* filter, UrlBuilderOptions options);

[VB] Function BuildUrl(ByVal owner As Control, ByVal initialUrl As String,

ByVal baseUrl As String, ByVal caption As String, ByVal filter As String, ByVal

options As UrlBuilderOptions) As String

[JScript] function BuildUrl(owner: Control, initialUrl: String, baseUrl: String,

caption: String, filter: String, options: UrlBuilderOptions): String;

3

4

5

6

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Launches an editor to build a URL property.

Return Value: A string that contains the URL returned by the

System.Web.UI.Design.UrlBuilder. The control that contains the property to build. The initial URL to display in the selection interface. The base URL to display in the selection interface. A caption that presents a message in the selection interface. A filter for the types of files listed in the selection interface.

The options for the System. Web. UI. Design. Url Builder to use.

IWebFormsDocumentService interface (System.Web.UI.Design)
BuildUrl

### Description

Provides methods to access services for tracking the loading state of a web forms document, handling events at load time, accessing a document's location, managing a document's undo service, and setting a new selection within the document.

DocumentUrl

BuildUrl

[C#] string DocumentUrl {get;}

[C++] String\* get DocumentUrl();

[VB] ReadOnly Property DocumentUrl As String

[JScript] abstract function get DocumentUrl(): String;

Gets or sets the URL at which the document is located.
IsLoading
BuildUrl
[C#] bool IsLoading {get;}
[C++] bool get_IsLoading();
[VB] ReadOnly Property IsLoading As Boolean
[JScript] abstract function get IsLoading(): Boolean;
Description
Indicates whether the document service is currently loading.
BuildUrl
[C#] event EventHandler LoadComplete;
[C++]event EventHandler* LoadComplete;
[VB] Event LoadComplete As EventHandler
Description
Occurs when the service has finished loading.
This event provides an opportunity to perform operations immediately after
loading has completed. Events that should occur at load time can be registered as
event handlers for this event.
CreateDiscardableUndoUnit
[C#] object CreateDiscardableUndoUnit();

1	[C++] Object* CreateDiscardableUndoUnit();
2	[VB] Function CreateDiscardableUndoUnit() As Object
3	[JScript] function CreateDiscardableUndoUnit(): Object;
4	
5	Description
6	Creates a discardable undo unit.
7	Return Value: The new discardable undo unit.
8	DiscardUndoUnit
9	
10	[C#] void DiscardUndoUnit(object discardableUndoUnit);
11	[C++] void DiscardUndoUnit(Object* discardableUndoUnit);
12	[VB] Sub DiscardUndoUnit(ByVal discardableUndoUnit As Object)
13	[JScript] function DiscardUndoUnit(discardableUndoUnit : Object);
14	
15	Description
16	Discards the specified undo unit. The undo unit to discard.
17	EnableUndo
18	
19	[C#] void EnableUndo(bool enable);
20	[C++] void EnableUndo(bool enable);
21	[VB] Sub EnableUndo(ByVal enable As Boolean)
22	[JScript] function EnableUndo(enable : Boolean);
23	
24	Description
25	

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Enables the ability to undo actions that occur within undoable action units or transactions. **true** if actions should be undoable; otherwise, **false**.

UpdateSelection

[C#] void UpdateSelection();

[C++] void UpdateSelection();

[VB] Sub UpdateSelection()

[JScript] function UpdateSelection();

# Description

When implemented in a derived class, updates the current selection.

When implemented in a derived class, this method updates the current selection.

ReadWriteControlDesigner class (System.Web.UI.Design)

UpdateSelection

# Description

Provides design-time functionality for read/write server controls.

ReadWriteControlDesigner

Example Syntax:

UpdateSelection

[C#] public ReadWriteControlDesigner();

[C++] public: ReadWriteControlDesigner();

1	[VB] Public Sub New()
2	[JScript] public function ReadWriteControlDesigner();
3	
4	Description
5	Initializes an instance of the
6	System.Web.UI.Design.ReadWriteControlDesigner class.
7	AllowResize
8	AssociatedComponents
9	Behavior
10	Component
11	DataBindings
12	DesignTimeElement
13	DesignTimeElementView
14	DesignTimeHtmlRequiresLoadComplete
15	ID
16	InheritanceAttribute
17	Inherited
18	IsDirty
19	ReadOnly
20	ShadowProperties
21	ShouldCodeSerialize
22	Verbs
23	MapPropertyToStyle
24	
25	[C#] protected virtual void MapPropertyToStyle(string propName, object

1	1
1	varPropValue);
2	[C++] protected: virtual void MapPropertyToStyle(String* propName, Object*
3	varPropValue);
4	[VB] Overridable Protected Sub MapPropertyToStyle(ByVal propName As
5	String, ByVal varPropValue As Object)
6	[JScript] protected function MapPropertyToStyle(propName : String,
7	varPropValue : Object);
8	
9	Description
10	Maps a property, including description and value, to an Interactive HTML
11	style.
12	Return Value: This method must be overridden in a derived class to implement the
13	designer. The name of the property to map. The value of the property.
14	OnBehaviorAttached
15	
16	[C#] protected override void OnBehaviorAttached();
17	[C++] protected: void OnBehaviorAttached();
18	[VB] Overrides Protected Sub OnBehaviorAttached()
19	[JScript] protected override function OnBehaviorAttached();
20	
21	Description
22	Provides notification that is raised upon a behavior being attached to the
23	designer.
24	OnComponentChanged
25	

1	
2	[C#] public override void OnComponentChanged(object sender,
3	ComponentChangedEventArgs ce);
4	[C++] public: void OnComponentChanged(Object* sender,
5	ComponentChangedEventArgs* ce);
6	[VB] Overrides Public Sub OnComponentChanged(ByVal sender As Object,
7	ByVal ce As ComponentChangedEventArgs)
8	[JScript] public override function OnComponentChanged(sender : Object, ce :
9	ComponentChangedEventArgs);
10	
11	Description
12	Represents the method that will handle the
- 1	
13	System.ComponentModel.Design.IComponentChangeService.ComponentCha
13	System.ComponentModel.Design.IComponentChangeService.ComponentChanged event of the
14	nged event of the
14	nged event of the  System.ComponentModel.Design.IComponentChangeService.
14 15	nged event of the  System.ComponentModel.Design.IComponentChangeService .  The
14 15 16 17	nged event of the  System.ComponentModel.Design.IComponentChangeService .  The  System.ComponentModel.Design.IComponentChangeService.ComponentCha
14   15   16   17   18	nged event of the  System.ComponentModel.Design.IComponentChangeService.  The  System.ComponentModel.Design.IComponentChangeService.ComponentChanged event occurs after a property has been changed. This delegate allows
14   15   16   17   18   19	nged event of the  System.ComponentModel.Design.IComponentChangeService.  The  System.ComponentModel.Design.IComponentChangeService.ComponentChanged event occurs after a property has been changed. This delegate allows implementors to do any post-processing that may be needed after a property
14 15 16 17 18 19 20	nged event of the  System.ComponentModel.Design.IComponentChangeService.  The  System.ComponentModel.Design.IComponentChangeService.ComponentChanged event occurs after a property has been changed. This delegate allows implementors to do any post-processing that may be needed after a property change. The object sending the event. The event object used when raising a
14   15   16   17   18   19   20   21	nged event of the  System.ComponentModel.Design.IComponentChangeService.  The  System.ComponentModel.Design.IComponentChangeService.ComponentChanged event occurs after a property has been changed. This delegate allows implementors to do any post-processing that may be needed after a property change. The object sending the event. The event object used when raising a System.ComponentModel.Design.IComponentChangeService.ComponentCha

	1	
	2	
	3	Description
	4	Provides design-time functionality for template-based server controls.
	5	TemplatedControlDesigner
	6	Example Syntax:
	7	UpdateDesignTimeHtml
	8	
	9	[C#] public TemplatedControlDesigner();
	10	[C++] public: TemplatedControlDesigner();
մում մում հոմ Գոտ հոմ Արժ Արժ դում	11	[VB] Public Sub New()
Half Ham	12	[JScript] public function TemplatedControlDesigner();
initi initi	13	
, (m),	14	Description
	15	Initializes a new instance of the
arraya	16	System.Web.UI.Design.TemplatedControlDesigner class.
•	17	ActiveTemplateEditingFrame
	18	UpdateDesignTimeHtml
	19	
	20	[C#] public ITemplateEditingFrame ActiveTemplateEditingFrame {get;}
	21	[C++] public:property ITemplateEditingFrame*
	22	get_ActiveTemplateEditingFrame();
	23	[VB] Public ReadOnly Property ActiveTemplateEditingFrame As
	24	ITemplateEditingFrame
	25	[JScript] public function get ActiveTemplateEditingFrame():

1	ITemplateEditingFrame;
2	
3	Description
4	Gets the active template editing frame.
5	AllowResize
6	AssociatedComponents
7	Behavior
8	CanEnterTemplateMode
9	UpdateDesignTimeHtml
10	
11	
12	Description
13	Indicates whether or not this designer will allow the viewing or editing of
14	templates. This property is read-only.
15	Component
16	DataBindings
17	DesignTimeElement
18	DesignTimeElementView
19	DesignTimeHtmlRequiresLoadComplete
20	HidePropertiesInTemplateMode
21	UpdateDesignTimeHtml
22	
23	
24	Description
25	

2

3

5

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Indicates whether the properties of the control will be hidden when the control is placed into template editing mode.

The **System.Web.UI.Design.ControlDesigner.ID** property is never hidden.

ID

InheritanceAttribute

Inherited

InTemplateMode

UpdateDesignTimeHtml

### Description

Indicates whether or not the designer document is in template mode.

A document is in template mode when a template is currently being viewed or edited in the forms designer.

**IsDirty** 

ReadOnly

ShadowProperties

ShouldCodeSerialize

Verbs

CreateTemplateEditingFrame

[C#] protected abstract ITemplateEditingFrame

 $Create Template Editing Frame (Template Editing Verb \ verb);$ 

[C++] protected: virtual ITemplateEditingFrame\*

1	CreateTemplateEditingFrame(TemplateEditingVerb* verb) = 0;
2	[VB] MustOverride Protected Function CreateTemplateEditingFrame(ByVal verb
3	As TemplateEditingVerb) As ITemplateEditingFrame
4	[JScript] protected abstract function CreateTemplateEditingFrame(verb:
5	TemplateEditingVerb): ITemplateEditingFrame;
6	
7	Description
8	Creates a template editing frame for the specified verb.
9	Return Value: The new template editing frame. The template editing verb to create
10	a template editing frame for.
11	Dispose
12	
13	[C#] protected override void Dispose(bool disposing);
14	[C++] protected: void Dispose(bool disposing);
15	[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)
16	[JScript] protected override function Dispose(disposing : Boolean);
17	
18	Description
19	Releases the unmanaged resources used by the
20	System.Web.UI.Design.TemplatedControlDesigner and optionally releases the
21	managed resources.
22	This method is called by the public <b>Dispose()</b> method and the
23	System.Object.Finalize method. true to release both managed and unmanaged
24	resources; false to release only unmanaged resources.
25	EnterTemplateMode
	2   3   4   5   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24

1	
2	[C#] public void EnterTemplateMode(ITemplateEditingFrame
3	newTemplateEditingFrame);
4	[C++] public: void EnterTemplateMode(ITemplateEditingFrame*
5	newTemplateEditingFrame);
6	[VB] Public Sub EnterTemplateMode(ByVal newTemplateEditingFrame As
7	ITemplateEditingFrame)
8	[JScript] public function EnterTemplateMode(newTemplateEditingFrame :
9	ITemplateEditingFrame);
10	
11	Description
12	Opens a particular template frame object for editing in the designer.
13	If already in template mode and editing a different template frame, this
14	method calls
15	System. Web. UI. Design. Templated Control Designer. Exit Template Mode (System) and the support of the control of the contr
16	m.Boolean,System.Boolean,System.Boolean) to save all its templates and close
17	that frame. The template editing frame object to open in the designer.
18	ExitTemplateMode
19	
20	[C#] public void ExitTemplateMode(bool fSwitchingTemplates, bool fNested,
21	bool fSave);
22	[C++] public: void ExitTemplateMode(bool fSwitchingTemplates, bool fNested,
23	bool fSave);
24	[VB] Public Sub ExitTemplateMode(ByVal fSwitchingTemplates As Boolean,
25	ByVal fNested As Boolean, ByVal fSave As Boolean)

[JScript] public function ExitTemplateMode(fSwitchingTemplates : Boolean, fNested: Boolean, fSave: Boolean); 2 3 Description Closes the currently active template editing frame after saving any relevant 5 changes. true when switching from one template editing frame to another, 6 otherwise false. true if this designer is nested (one or more levels) within another control whose designer is also in template editing mode; otherwise false. true if 8 templates shold be saved on exit; otherwise, false. GetCachedTemplateEditingVerbs 10 11 [C#] protected abstract TemplateEditingVerb[] 12 GetCachedTemplateEditingVerbs(); 13 [C++] protected: virtual TemplateEditingVerb\* 14 GetCachedTemplateEditingVerbs() [] = 0; 15 [VB] MustOverride Protected Function GetCachedTemplateEditingVerbs() As 16 TemplateEditingVerb() 17 [JScript] protected abstract function GetCachedTemplateEditingVerbs(): 18 TemplateEditingVerb[]; 19 20 Description 21 Gets the cached template editing verbs. 22 Return Value: An array of System. Web. UI. Design. Template Editing Verb 23 objects, if any. 24

**GetPersistInnerHtml** 

[C#] public override string GetPersistInnerHtml();
[C++] public: String* GetPersistInnerHtml();
[VB] Overrides Public Function GetPersistInnerHtml() As String
[JScript] public override function GetPersistInnerHtml(): String;
Description
Gets the HTML to be persisted for the content present within the associated
server control runtime.
Return Value: The persistable inner HTML.
GetTemplateContainerDataItemProperty
[C#] public virtual string GetTemplateContainerDataItemProperty(string
templateName);
[C++] public: virtual String* GetTemplateContainerDataItemProperty(String*
templateName);
[VB] Overridable Public Function GetTemplateContainerDataItemProperty(ByVal
templateName As String) As String
[JScript] public function GetTemplateContainerDataItemProperty(templateName:
String): String;
Description
Gets the data item property of the template's container.
Return Value: A string representing the data. The name of the template.
GetTemplateContainerDataSource

2 100
J
= ===
al.
alle.

11

12

13

14

15

16

17

18

19

20

21

23

24

[C#] public virtual IEnumerable GetTemplateContainerDataSource(string templateName);

[C++] public: virtual IEnumerable\* GetTemplateContainerDataSource(String\* templateName);

[VB] Overridable Public Function GetTemplateContainerDataSource(ByVal templateName As String) As IEnumerable

[JScript] public function GetTemplateContainerDataSource(templateName :

String): IEnumerable;

### Description

Gets the data source of the template's container.

Return Value: The data source of the container of the specified template. The name of the template.

GetTemplateContent

[C#] public abstract string GetTemplateContent(ITemplateEditingFrame editingFrame, string templateName, out bool allowEditing);

[C++] public: virtual String\* GetTemplateContent(ITemplateEditingFrame\* editingFrame, String\* templateName, bool\* allowEditing) = 0;

[VB] MustOverride Public Function GetTemplateContent(ByVal editingFrame As ITemplateEditingFrame, ByVal templateName As String, ByRef allowEditing As Boolean) As String

[JScript] public abstract function GetTemplateContent(editingFrame :

ITemplateEditingFrame, templateName: String, allowEditing: Boolean): String;

1335 lee@hayes pilc 509+324+9256 MS1-863US.APP

Į,		
1		
į		
dress times that if		

1	
2	Description
3	Gets the template's content.
4	Return Value: The content of the template. The template editing frame to retrieve
5	the content of. The name of the template. A boolean variable that will be set to
6	true if the template's content can be edited, or false if the content is read-only.
7	GetTemplateEditingVerbs
8	
9	[C#] public TemplateEditingVerb[] GetTemplateEditingVerbs();
10	[C++] public: TemplateEditingVerb* GetTemplateEditingVerbs() [];
11	[VB] Public Function GetTemplateEditingVerbs() As TemplateEditingVerb()
12	[JScript] public function GetTemplateEditingVerbs() : TemplateEditingVerb[];
13	
14	Description
15	Gets the template editing verbs available to the designer.
16	Return Value: The template editing verbs available to the designer.
17	GetTemplateFromText
18	
19	[C#] protected ITemplate GetTemplateFromText(string text);
20	[C++] protected: ITemplate* GetTemplateFromText(String* text);
21	[VB] Protected Function GetTemplateFromText(ByVal text As String) As
22	ITemplate
23	[JScript] protected function GetTemplateFromText(text : String) : ITemplate;
24	
25	Description

Creates a template from the specified text.

Return Value: An System.Web.UI.ITemplate from the specified text. The text to retrieve a template from.

**GetTemplatePropertyParentType** 

[C#] public virtual Type GetTemplatePropertyParentType(string templateName);

[C++] public: virtual Type\* GetTemplatePropertyParentType(String\* templateName);

[VB] Overridable Public Function GetTemplatePropertyParentType(ByVal templateName As String) As Type

[JScript] public function GetTemplatePropertyParentType(templateName : String)
: Type;

## Description

Gets the type of the parent of the template property.

Return Value: The type of the object that has the template property. The name of the template to return the type of the parent for.

GetTextFromTemplate

[C#] protected string GetTextFromTemplate(ITemplate template);

[C++] protected: String\* GetTextFromTemplate(ITemplate\* template);

[VB] Protected Function GetTextFromTemplate(ByVal template As ITemplate)

As String

25

[JScript] protected function GetTextFromTemplate(template : ITemplate) : String;

1	
2	Description
3	Gets a string of text that represents the specified template.
4	Return Value: A string that represents the specified template. The
5	System.Web.UI.ITemplate to convert to text.
6	OnBehaviorAttached
7	
8	[C#] protected override void OnBehaviorAttached();
9	[C++] protected: void OnBehaviorAttached();
10	[VB] Overrides Protected Sub OnBehaviorAttached()
11	[JScript] protected override function OnBehaviorAttached();
12	
13	Description
14	Notification that is called when the behavior is attached to the designer.
15	OnComponentChanged
16	
17	[C#] public override void OnComponentChanged(object sender,
18	ComponentChangedEventArgs ce);
19	[C++] public: void OnComponentChanged(Object* sender,
20	ComponentChangedEventArgs* ce);
21	[VB] Overrides Public Sub OnComponentChanged(ByVal sender As Object,
22	ByVal ce As ComponentChangedEventArgs)
23	[JScript] public override function OnComponentChanged(sender : Object, ce :
24	ComponentChangedEventArgs);
25	

٠,		
į		
their time their		
į		
ì		
į		

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Desc	ription

Delegate to handle the component changed event.

This is called after a property has been changed. It allows the implementor to do any post-processing that may be needed after a property change. The object sending the event. A

**System.ComponentModel.Design.ComponentChangedEventArgs** that provides data for the event.

OnSetParent

[C#] public override void OnSetParent();

[C++] public: void OnSetParent();

[VB] Overrides Public Sub OnSetParent()

[JScript] public override function OnSetParent();

### Description

Notification that is called when the associated control is parented.

OnTemplateModeChanged

[C#] protected virtual void OnTemplateModeChanged();

[C++] protected: virtual void OnTemplateModeChanged();

[VB] Overridable Protected Sub OnTemplateModeChanged()

[JScript] protected function OnTemplateModeChanged();

Provides notification and handles processing that should occur when the template mode is changed.

**PreFilterProperties** 

[C#] protected override void PreFilterProperties(IDictionary properties);

[C++] protected: void PreFilterProperties(IDictionary\* properties);

[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As

IDictionary)

2

3

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

[JScript] protected override function PreFilterProperties(properties: IDictionary);

### Description

Allows a designer to filter the set of member attributes the component it is designing will expose through a **System.ComponentModel.TypeDescriptor** object.

Return Value: The augmented set of attributes. If the method does not modify any attributes, it may just return a reference to its input parameter. If you do make a change to the attributes, you must create a new array.

This method is called immediately before its corresponding "Post" method. If you are overriding this method you should call the base implementation before you perform your own filtering. The member attributes for component.

SaveActiveTemplateEditingFrame

[C#] protected void SaveActiveTemplateEditingFrame();

[C++] protected: void SaveActiveTemplateEditingFrame();

[VB] Protected Sub SaveActiveTemplateEditingFrame()

[JScript] protected function SaveActiveTemplateEditingFrame(); 2 Description Saves the active template editing frame. SetTemplateContent 6 [C#] public abstract void SetTemplateContent(ITemplateEditingFrame editingFrame, string templateName, string templateContent); 8 [C++] public: virtual void SetTemplateContent(ITemplateEditingFrame\* 9 editingFrame, String\* templateName, String\* templateContent) = 0; 10 [VB] MustOverride Public Sub SetTemplateContent(ByVal editingFrame As 11 ITemplateEditingFrame, ByVal templateName As String, ByVal templateContent 12 As String) 13 [JScript] public abstract function SetTemplateContent(editingFrame: ITemplateEditingFrame, templateName: String, templateContent: String); 15 16 Description 17 Sets the specified template's content to the specified content. The template 18 editing frame to provide content for. The name of the template. The content to set for the template. 20 **UpdateDesignTimeHtml** 21 22 [C#] public override void UpdateDesignTimeHtml(); 23 [C++] public: void UpdateDesignTimeHtml(); [VB] Overrides Public Sub UpdateDesignTimeHtml()

1	[JScript] public override function UpdateDesignTimeHtml();
2	
3	Description
4	Updates the design-time HTML.
5	This method should be called to refresh the design-time display if the
6	template frames have been changed without calling methods that update the
7	design-time HTML automatically.
8	TemplateEditingService class (System.Web.UI.Design)
9	UpdateDesignTimeHtml
10	TemplateEditingService
11	Example Syntax:
12	UpdateDesignTimeHtml
13	SupportsNestedTemplateEditing
14	UpdateDesignTimeHtml
15	CreateFrame
16	
17	[C#] public ITemplateEditingFrame CreateFrame(TemplatedControlDesigner
18	designer, string frameName, string[] templateNames);
19	[C++] public:sealed ITemplateEditingFrame*
20	CreateFrame(TemplatedControlDesigner* designer, String* frameName, String*
21	templateNamesgc[]);
22	[VB] NotOverridable Public Function CreateFrame(ByVal designer As
23	TemplatedControlDesigner, ByVal frameName As String, ByVal
24	templateNames() As String) As ITemplateEditingFrame
25	

```
[JScript] public function CreateFrame(designer: TemplatedControlDesigner,
    frameName: String, templateNames: String[]): ITemplateEditingFrame;
2
           CreateFrame
3
4
    [C#] public ITemplateEditingFrame CreateFrame(TemplatedControlDesigner
5
    designer, string frameName, string[] templateNames, Style controlStyle, Style[]
6
    templateStyles);
    [C++] public: sealed ITemplateEditingFrame*
8
    CreateFrame(TemplatedControlDesigner* designer, String* frameName, String*
    templateNames __gc[], Style* controlStyle, Style* templateStyles[]);
10
    [VB] NotOverridable Public Function CreateFrame(ByVal designer As
11
    TemplatedControlDesigner, ByVal frameName As String, ByVal
12
    templateNames() As String, ByVal controlStyle As Style, ByVal templateStyles()
13
    As Style) As ITemplateEditingFrame
14
    [JScript] public function CreateFrame(designer: TemplatedControlDesigner,
15
    frameName: String, templateNames: String[], controlStyle: Style, templateStyles
16
    : Style[]) : ITemplateEditingFrame;
17
           Dispose
18
19
    [C#] public void Dispose();
20
    [C++] public: sealed void Dispose();
21
    [VB] NotOverridable Public Sub Dispose()
22
    [JScript] public function Dispose();
23
           Finalize
24
25
```

1	
2	[C#] ~TemplateEditingService();
3	[C++] ~TemplateEditingService();
4	[VB] Overrides Protected Sub Finalize()
5	[JScript] protected override function Finalize();
6	GetContainingTemplateName
7	
8	[C#] public string GetContainingTemplateName(Control control);
9	[C++] public:sealed String* GetContainingTemplateName(Control* control);
10	[VB] NotOverridable Public Function GetContainingTemplateName(ByVal
11	control As Control) As String
12	[JScript] public function GetContainingTemplateName(control : Control) : String;
13	TemplateEditingVerb class (System.Web.UI.Design)
14	ToString
15	
16	
17	Description
18	Represents a designer verb that creates a template editing frame, and can
19	only be invoked by a template editor.
20	TemplateEditingVerb
21	Example Syntax:
22	ToString
23	
24	[C#] public TemplateEditingVerb(string text, int index,
25	TemplatedControlDesigner designer);

	1	[C++] public: TemplateEditingVerb(String* text, int index,
	2	TemplatedControlDesigner* designer);
	3	[VB] Public Sub New(ByVal text As String, ByVal index As Integer, ByVal
	4	designer As TemplatedControlDesigner)
	5	[JScript] public function TemplateEditingVerb(text : String, index : int, designer :
	6	TemplatedControlDesigner);
	7	
	8	Description
	9	Initializes a new instance of the
	10	System.Web.UI.Design.TemplateEditingVerb class. The text to show for this
	11	verb on a menu. An optional integer value that can be used by a designer; typically
	12	to indicate the index of the verb within a set of verbs. The designer that can use
	13	this verb.
	14	Checked
the tent and the form that	15	CommandID
	16	Enabled
ii.	17	Index
	18	ToString
	19	
	20	
	21	Description
	22	Gets or sets the index, or other user data, for this verb.
	23	This property is sometimes used to store the index of each verb in a set of
	24	verbs. When implementing a custom designer, this property can be used to store
	25	integer data in a custom defined manner.

typically

OleStatus Supported 2 Text 3 Visible 4 Dispose 5 6 [C#] public void Dispose(); 7 [C++] public: \_sealed void Dispose(); 8 [VB] NotOverridable Public Sub Dispose() 9 [JScript] public function Dispose(); Releases all resources used by the 10 System.Web.UI.Design.TemplateEditingVerb. 11 12 Description 13 Releases all resources used by the 14 System.Web.UI.Design.TemplateEditingVerb. 15 Calling System.Web.UI.Design.TemplateEditingVerb.Dispose allows 16 the resources used by the System. Web. UI. Design. Template Editing Verb to be 17 reallocated for other purposes. For more information about 18  $System. Web. UI. Design. Template Editing Verb. Dispose\ ,\ see\ .$ 19 Dispose 20 21 [C#] protected virtual void Dispose(bool disposing); 22 [C++] protected: virtual void Dispose(bool disposing); 23 [VB] Overridable Protected Sub Dispose(ByVal disposing As Boolean) 24 [JScript] protected function Dispose(disposing : Boolean); 25

Description 2 Releases the unmanaged resources used by the 3 System.Web.UI.Design.TemplateEditingVerb and optionally releases the managed resources. 5 This method is called by the public **Dispose()** method and the 6 System.Object.Finalize method. true to release both managed and unmanaged resources; false to release only unmanaged resources. 8 **Finalize** 9 10 [C#] ~TemplateEditingVerb(); 11 [C++] ~TemplateEditingVerb(); 12 [VB] Overrides Protected Sub Finalize() 13 [JScript] protected override function Finalize(); 14 TextControlDesigner class (System.Web.UI.Design) 15 **ToString** 16 17 18 Description 19 Provides design-time support for the ASP.NET 20 System.Web.UI.WebControls.Label and System.Windows.Forms.HyperLink 21 server controls. 22 TextControlDesigner 23 Example Syntax: 24 **ToString** 25

1	
2	[C#] public TextControlDesigner();
3	[C++] public: TextControlDesigner();
4	[VB] Public Sub New()
5	[JScript] public function TextControlDesigner();
6	AllowResize
7	AssociatedComponents
8	Behavior
9	Component
10	DataBindings
11	DesignTimeElement
12	DesignTimeElementView
13	DesignTimeHtmlRequiresLoadComplete
14	ID
15	InheritanceAttribute
16	Inherited
17	IsDirty
18	ReadOnly
19	ShadowProperties
20	ShouldCodeSerialize
21	Verbs
22	GetDesignTimeHtml

1	[VB] Overrides Public Function GetDesignTimeHtml() As String
2	[JScript] public override function GetDesignTimeHtml(): String;
3	
4	Description
5	Gets the HTML that is used to represent the associated control at design
6	time.
7	Return Value: The HTML used to represent the control at design-time.
8	GetPersistInnerHtml
9	
10	[C#] public override string GetPersistInnerHtml();
11	[C++] public: String* GetPersistInnerHtml();
12	[VB] Overrides Public Function GetPersistInnerHtml() As String
13	[JScript] public override function GetPersistInnerHtml(): String;
14	
15	Description
16	Gets the HTML to persist for the inner tags of the control.
17	Return Value: The HTML for the tags of the control.
18	Initialize
19	
20	[C#] public override void Initialize(IComponent component);
21	[C++] public: void Initialize(IComponent* component);
22	[VB] Overrides Public Sub Initialize(ByVal component As IComponent)
23	[JScript] public override function Initialize(component : IComponent);
24	
25	Description

Initializes this designer with the specified component. The component for 1 this designer to design. 2 TextDataBindingHandler class (System.Web.UI.Design) UpdateDesignTimeHtml Description Provides a data binding handler for a text property of a control. TextDataBindingHandler Example Syntax: UpdateDesignTimeHtml [C#] public TextDataBindingHandler(); [C++] public: TextDataBindingHandler(); [VB] Public Sub New() [JScript] public function TextDataBindingHandler(); DataBindControl [C#] public override void DataBindControl(IDesignerHost designerHost, Control control); [C++] public: void DataBindControl(IDesignerHost\* designerHost, Control\* 21 control); 22 [VB] Overrides Public Sub DataBindControl(ByVal designerHost As 23 IDesignerHost, ByVal control As Control) 24

[JScript] public override function DataBindControl(designerHost: IDesignerHost,

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

filter As String) As String

control: Control); Description Adds this data binding to the specified control. The designer host for the document that contains the control. The control to add this data binding to. UrlBuilder class (System.Web.UI.Design) **ToString** Description Launches a URL editor that allows a user to select a URL. The System.Web.UI.Design.UrlBuilder.BuildUrl(System.ComponentModel.ICom ponent, System. Windows. Forms. Control, System. String, System. String, System. String) method launches a user interface for selecting a URL. BuildUrl [C#] public static string BuildUrl(IComponent component, Control owner, string initialUrl, string caption, string filter); [C++] public: static String\* BuildUrl(IComponent\* component, Control\* owner, String\* initialUrl, String\* caption, String\* filter); [VB] Public Shared Function BuildUrl(ByVal component As IComponent, ByVal owner As Control, ByVal initialUrl As String, ByVal caption As String, ByVal

[JScript] public static function BuildUrl(component : IComponent, owner :

Control, initialUrl: String, caption: String, filter: String): String; Launches the Url Picker to build a Url property.

## Description

Launches the Url Picker to build a Url property.

Return Value: The Url returned from the Url Picker. The component whose site is to be used to access design-time services. The control used to parent the picker window. The initial Url to be shown in the picker window. The caption of the picker window. The filter for selecting files in the picker window.

#### BuildUrl

[C#] public static string BuildUrl(IComponent component, Control owner, string initialUrl, string caption, string filter, UrlBuilderOptions options);
[C++] public: static String\* BuildUrl(IComponent\* component, Control\* owner, String\* initialUrl, String\* caption, String\* filter, UrlBuilderOptions options);
[VB] Public Shared Function BuildUrl(ByVal component As IComponent, ByVal owner As Control, ByVal initialUrl As String, ByVal caption As String, ByVal filter As String, ByVal options As UrlBuilderOptions) As String
[JScript] public static function BuildUrl(component: IComponent, owner:
Control, initialUrl: String, caption: String, filter: String, options:
UrlBuilderOptions): String;

#### Description

Launches the Url Picker to build a Url property using the specified System.Web.UI.Design.UrlBuilderOptions .

Return Value: The Url returned from the Url Picker. The component whose site is
to be used to access design-time services. The control used to parent the picker
window. The initial Url to be shown in the picker window. The caption of the
picker window. The filter for selecting files in the picker window. A
System.Web.UI.Design.UrlBuilderOptions indicating the options for Url
selection.

UrlBuilderOptions enumeration (System.Web.UI.Design)
ToString

## Description

Defines identifiers for settings for a **System.Web.UI.Design.UrlBuilder** . ToString

[C#] public const UrlBuilderOptions NoAbsolute;

[C++] public: const UrlBuilderOptions NoAbsolute;

[VB] Public Const NoAbsolute As UrlBuilderOptions

[JScript] public var NoAbsolute : UrlBuilderOptions;

# Description

Build a URL that references a path relative to the current path, rather than one that consists of a fully qualified, absolute path reference URL.

**ToString** 

[C#] public const UrlBuilderOptions None;

```
[C++] public: const UrlBuilderOptions None;
    [VB] Public Const None As UrlBuilderOptions
    [JScript] public var None : UrlBuilderOptions;
3
    Description
5
           Use no additional options for the System.Web.UI.Design.UrlBuilder .
6
           UrlEditor class (System.Web.UI.Design)
           ToString
8
9
10
    Description
11
           Provides a user interface for selecting a URL.
12
           UrlEditor
13
           Example Syntax:
14
           ToString
15
16
     [C#] public UrlEditor();
17
     [C++] public: UrlEditor();
18
     [VB] Public Sub New()
19
     [JScript] public function UrlEditor();
20
            Caption
21
            ToString
22
23
     [C#] protected virtual string Caption {get;}
     [C++] protected: __property virtual String* get_Caption();
```

	1	[VB] Overridable Protected ReadOnly Property Caption As String
	2	[JScript] protected function get Caption(): String;
	3	
	4	Description
	5	Gets the caption for the URL.
	6	Filter
	7	ToString
	8	
	9	[C#] protected virtual string Filter {get;}
the stead	10	[C++] protected:property virtual String* get_Filter();
	11	[VB] Overridable Protected ReadOnly Property Filter As String
	12	[JScript] protected function get Filter(): String;
	13	
	14	Description
	15	Gets or sets the file extensions to filter the file list for.
	16	Options
	17	ToString
	18	
	19	[C#] protected virtual UrlBuilderOptions Options {get;}
	20	[C++] protected:property virtual UrlBuilderOptions get_Options();
	21	[VB] Overridable Protected ReadOnly Property Options As UrlBuilderOptions
	22	[JScript] protected function get Options(): UrlBuilderOptions;
	23	
	24	Description
	25	Gets the options for a URL builder to use.

$\mathbf{T}^{2}$	.11	4.	7.	1_	
r.	a	T.	Vа	ш	пе

3

4

5

6 7

8

9

10

11

12

13

14

18

17

16

20

21

19

2223

24

[C#] public override object EditValue(ITypeDescriptorContext context,

IServiceProvider provider, object value);

[C++] public: Object\* EditValue(ITypeDescriptorContext\* context,

IServiceProvider\* provider, Object\* value);

[VB] Overrides Public Function EditValue(ByVal context As

ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As

Object) As Object

[JScript] public override function EditValue(context: ITypeDescriptorContext,

provider: IServiceProvider, value: Object;

Description

Edits the value of the specified object using the editor style provided by the System.Web.UI.Design.UrlEditor.GetEditStyle(System.ComponentModel.ITy peDescriptorContext) method.

Return Value: The new value of the object. If the value of the object hasn't changed, this should return the same object it was passed.

A service provider is provided so that any required editing services can be obtained. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A service provider object through which editing services may be obtained. An instance of the value being edited.

GetEditStyle

[C#] public override UITypeEditorEditStyle

1	GetEditStyle(ITypeDescriptorContext context);
2	[C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext*
3	context);
4	[VB] Overrides Public Function GetEditStyle(ByVal context As
5	ITypeDescriptorContext) As UITypeEditorEditStyle
6	[JScript] public override function GetEditStyle(context : ITypeDescriptorContext)
7	: UITypeEditorEditStyle;
8	
9	Description
10	Gets the editing style of the
11	System.Web.UI.Design.UrlEditor.EditValue(System.ComponentModel.IType
12	DescriptorContext,System.IServiceProvider,System.Object) method.
13	Return Value: One of the System.Drawing.Design.UITypeEditorEditStyle
14	values indicating the provided editing style. If the method is not supported, this
15	will return System.Drawing.Design.UITypeEditorEditStyle.None . An
16	System.ComponentModel.ITypeDescriptorContext that can be used to gain
17	additional context information.
18	UserControlDesigner class (System.Web.UI.Design)
19	ToString
20	
21	
22	Description
23	Provides design-time support for web user controls.
24	UserControlDesigner
25	Example Syntax:

	1	ToString
	2	
	3	[C#] public UserControlDesigner();
	4	[C++] public: UserControlDesigner();
	5	[VB] Public Sub New()
	6	[JScript] public function UserControlDesigner();
	7	
	8	Description
	9	Initializes a new instance of the
# # # # T	10	System.Web.UI.Design.UserControlDesigner class.
The said than that that the that	11	AllowResize
and thus	12	ToString
	13	
	14	[C#] public override bool AllowResize {get;}
	15	[C++] public:property virtual bool get_AllowResize();
	16	[VB] Overrides Public ReadOnly Property AllowResize As Boolean
	17	[JScript] public function get AllowResize(): Boolean;
	18	
	19	Description
	20	Indicates whether all user controls are resizable.
	21	AssociatedComponents
	22	Behavior
	23	Component
	24	DataBindings
	25	DesignTimeElement

1	DesignTimeElementView
2	DesignTimeHtmlRequiresLoadComplete
3	ID
4	InheritanceAttribute
5	Inherited
6	IsDirty
7	ReadOnly
8	ShadowProperties
9	ShouldCodeSerialize
10	ToString
11	
12	
13	Description
14	Indicates whether this designer should add a field declaration for the
15	control in the code-behind file for the current design document.
16	Verbs
17	GetDesignTimeHtml
18	
19	[C#] public override string GetDesignTimeHtml();
20	[C++] public: String* GetDesignTimeHtml();
21	[VB] Overrides Public Function GetDesignTimeHtml() As String
22	[JScript] public override function GetDesignTimeHtml(): String;
23	
24	Description
25	

1	Gets the HTML that is used to represent the control at design time.
2	Return Value: The HTML that is used to represent the control at design time.
3	GetPersistInnerHtml
4	
5	[C#] public override string GetPersistInnerHtml();
6	[C++] public: String* GetPersistInnerHtml();
7	[VB] Overrides Public Function GetPersistInnerHtml() As String
8	[JScript] public override function GetPersistInnerHtml(): String;
9	
10	Description
11	Gets the inner HTML that is persisted for the control.
12	Return Value: The content to be persisted from within the control's tags.
13	WebControlToolboxItem class (System.Web.UI.Design)
14	UpdateDesignTimeHtml
15	
16	
17	Description
18	Provides a base class for a Web server control
19	System.Drawing.Design.ToolboxItem .
20	WebControlToolboxItem
21	Example Syntax:
22	UpdateDesignTimeHtml
23	
24	[C#] public WebControlToolboxItem();
25	[C++] public: WebControlToolboxItem();

1	[VB] Public Sub New()
2	[JScript] public function WebControlToolboxItem(); Initializes a new instance of
3	the System. Web. UI. Design. WebControl Toolbox I tem class.
4	
5	Description
6	Initializes a new instance of the
7	System.Web.UI.Design.WebControlToolboxItem class.
8	The display name will be a short version of the type name.
9	WebControlToolboxItem
10	Example Syntax:
11	UpdateDesignTimeHtml
12	
13	[C#] public WebControlToolboxItem(Type type);
14	[C++] public: WebControlToolboxItem(Type* type);
15	[VB] Public Sub New(ByVal type As Type)
16	[JScript] public function WebControlToolboxItem(type: Type);
17	
18	Description
19	Initializes a new instance of the
20	System.Web.UI.Design.WebControlToolboxItem class. The fully qualified type
21	name of the tool for this toolbox item
22	AssemblyName
23	Bitmap
24	DisplayName
25	Filter

	2	TypeName
	3	CreateComponentsCore
	4	
	5	[C#] protected override IComponent[] CreateComponentsCore(IDesignerHost
	6	host);
	7	[C++] protected: IComponent* CreateComponentsCore(IDesignerHost* host) [];
	8	[VB] Overrides Protected Function CreateComponentsCore(ByVal host As
	9	IDesignerHost) As IComponent()
	10	[JScript] protected override function CreateComponentsCore(host:
	11	IDesignerHost): IComponent[];
	12	
	13	Description
	14	Creates objects from each Type contained in this
	15	System.Drawing.Design.ToolboxItem, and adds them to the specified designer.
	16	Return Value: An array of created System.ComponentModel.IComponent
	17	objects.
	18	You can override this method to alter the way components are created. The
	19	System.ComponentModel.Design.IDesignerHost to use to create the
	20	components.
	21	Deserialize
	22	
	23	[C#] protected override void Deserialize(SerializationInfo info, StreamingContext
	24	context);

Locked

[C++] protected: void Deserialize(SerializationInfo\* info, StreamingContext

1	context);
2	[VB] Overrides Protected Sub Deserialize(ByVal info As SerializationInfo, ByVal
3	context As StreamingContext)
4	[JScript] protected override function Deserialize(info : SerializationInfo, context :
5	StreamingContext);
6	
7	Description
8	
9	GetToolAttributeValue
10	
11	[C#] public object GetToolAttributeValue(IDesignerHost host, Type
12	attributeType);
13	[C++] public: Object* GetToolAttributeValue(IDesignerHost* host, Type*
14	attributeType);
15	[VB] Public Function GetToolAttributeValue(ByVal host As IDesignerHost,
16	ByVal attributeType As Type) As Object
17	[JScript] public function GetToolAttributeValue(host: IDesignerHost,
18	attributeType: Type): Object;
19	
20	Description
21	Gets the value of the attribute of the specified type of the toolbox item.
22	Return Value: The value of the specified type of attribute.
23	The default implementation will throw an ArgumentException if the
24	attributeType parameter is anything other than a PersistChildrenAttribute. The
25	

	1	designer host for the current design document. The type of attribute to retrieve				
	2	value of.				
	3	GetToolHtml				
	4					
	5	[C#] public string GetToolHtml(IDesignerHost host);				
	6	[C++] public: String* GetToolHtml(IDesignerHost* host);				
	7	[VB] Public Function GetToolHtml(ByVal host As IDesignerHost) As String				
	8	[JScript] public function GetToolHtml(host : IDesignerHost) : String;				
	9					
	10	Description				
	11	Gets the HTML associated with the tool.				
	12	Return Value: The HTML associated with the tool. The designer host for the				
	13	current design document.				
	14	GetToolType				
	15					
	16	[C#] public Type GetToolType(IDesignerHost host);				
5 <b>:</b> \$	17	[C++] public: Type* GetToolType(IDesignerHost* host);				
	18	[VB] Public Function GetToolType(ByVal host As IDesignerHost) As Type				
	19	[JScript] public function GetToolType(host : IDesignerHost) : Type;				
	20					
	21	Description				
	22	Gets the type of this toolbox item.				
	23	Return Value: The type of this toolbox item. The designer host for the current				
	24	design document.				
	25	Initialize				

1	
1	
2	[C#] public override void Initialize(Type type);
3	[C++] public: void Initialize(Type* type);
4	[VB] Overrides Public Sub Initialize(ByVal type As Type)
5	[JScript] public override function Initialize(type: Type);
6	
7	Description
8	Initializes this toolbox item. The type of this Web server control toolbox
9	item.
10	Serialize
11	
12	[C#] protected override void Serialize(SerializationInfo info, StreamingContext
13	context);
14	[C++] protected: void Serialize(SerializationInfo* info, StreamingContext
15	context);
16	[VB] Overrides Protected Sub Serialize(ByVal info As SerializationInfo, ByVal
17	context As StreamingContext)
18	[JScript] protected override function Serialize(info : SerializationInfo, context :
19	StreamingContext);
20	
21	Description
22	
23	XmlFileEditor class (System.Web.UI.Design)
24	ToString
25	

	1	
	2	
	3	Description
	4	Provides a user interface for selecting an XML File.
	5	XmlFileEditor
	6	Example Syntax:
	7	ToString
	8	
	9	[C#] public XmlFileEditor();
Ame Ame	10	[C++] public: XmlFileEditor();
was had had had	11	[VB] Public Sub New()
	12	[JScript] public function XmlFileEditor();
	13	EditValue
	14	
	15	[C#] public override object EditValue(ITypeDescriptorContext context,
	16	IServiceProvider provider, object value);
	17	[C++] public: Object* EditValue(ITypeDescriptorContext* context,
	18	IServiceProvider* provider, Object* value);
	19	[VB] Overrides Public Function EditValue(ByVal context As
	20	ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As
	21	Object) As Object
	22	[JScript] public override function EditValue(context : ITypeDescriptorContext,
	23	provider : IServiceProvider, value : Object) : Object;
	24	
	25	Description

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Edits the value of the specified object using the specified service provider and context.

Return Value: The new value of the object. If the value of the object hasn't changed, this method should return the same object it was passed.

A service provider is provided so that any required editing services can be obtained. An **System.ComponentModel.ITypeDescriptorContext** that can be used to gain additional context information. A service provider object through which editing services may be obtained. An instance of the value being edited.

GetEditStyle

[C#] public override UITypeEditorEditStyle

GetEditStyle(ITypeDescriptorContext context);

[C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetEditStyle(ByVal context As

ITypeDescriptorContext) As UITypeEditorEditStyle

[JScript] public override function GetEditStyle(context : ITypeDescriptorContext)

: UITypeEditorEditStyle;

### Description

Gets the editor style used by the

System.Web.UI.Design.XmlFileEditor.EditValue(System.ComponentModel.I TypeDescriptorContext,System.IServiceProvider,System.Object) method.

Return Value: An value indicating the provided editing style. An

```
System.ComponentModel.ITypeDescriptorContext that can be used to gain
    additional context information.
2
           XmlUrlEditor class (System.Web.UI.Design)
           ToString
6
    Description
7
           Provides a user interface for editing an URL embedded in XML.
           XmlUrlEditor
           Example Syntax:
10
           ToString
11
12
    [C#] public XmlUrlEditor();
13
    [C++] public: XmlUrlEditor();
14
    [VB] Public Sub New()
15
    [JScript] public function XmlUrlEditor();
16
           Caption
17
           ToString
18
19
    [C#] protected override string Caption {get;}
20
    [C++] protected: __property virtual String* get Caption();
21
    [VB] Overrides Protected ReadOnly Property Caption As String
22
    [JScript] protected function get Caption(): String;
23
24
    Description
```

1	Gets or sets the caption for the URL.
2	Filter
3	ToString
4	
5	[C#] protected override string Filter {get;}
6	[C++] protected:property virtual String* get_Filter();
7	[VB] Overrides Protected ReadOnly Property Filter As String
8	[JScript] protected function get Filter(): String;
9	
10	Description
11	Gets or sets the file extensions to filter the file list for.
12	Options
13	ToString
14	
15	[C#] protected override UrlBuilderOptions Options {get;}
16	[C++] protected:property virtual UrlBuilderOptions get_Options();
17	[VB] Overrides Protected ReadOnly Property Options As UrlBuilderOptions
18	[JScript] protected function get Options(): UrlBuilderOptions;
19	
20	Description
21	Gets the options for a URL builder to use.
22	XslUrlEditor class (System.Web.UI.Design)
23	ToString
24	
25	

```
Description
           Provides a user interface for selecting an XSL file.
           XslUrlEditor
           Example Syntax:
           ToString
    [C#] public XslUrlEditor();
    [C++] public: XslUrlEditor();
10
    [VB] Public Sub New()
11
    [JScript] public function XslUrlEditor();
           Caption
13
           ToString
14
15
    [C#] protected override string Caption {get;}
    [C++] protected: property virtual String* get Caption();
    [VB] Overrides Protected ReadOnly Property Caption As String
18
    [JScript] protected function get Caption(): String;
19
20
    Description
21
           Gets or sets the caption for the URL.
22
           Filter
23
           ToString
24
25
```

25

[C#] protected override string Filter {get;}
[C++] protected:property virtual String* get_Filter();
[VB] Overrides Protected ReadOnly Property Filter As String
[JScript] protected function get Filter(): String;

# System.Web.UI,Design.WebControls

Description

Description

The **System.Web.UI.Design.WebControls** namespace contains classes that can be used to extend design-time support for Web server controls.

AdRotatorDesigner class (System.Web.UI.Design.WebControls)

Description

Provides design-time support for the

System.Web.UI.WebControls.AdRotator Web server control.

Constructors:

AdRotatorDesigner

Example Syntax:

	1
1	
2	[C#] public AdRotatorDesigner();
3	[C++] public: AdRotatorDesigner();
4	[VB] Public Sub New()
5	[JScript] public function AdRotatorDesigner();
6	Properties:
7	AllowResize
8	AssociatedComponents
9	Behavior
 10	Component
11	DataBindings
12	DesignTimeElement
13	DesignTimeElementView
14	DesignTimeHtmlRequiresLoadComplete
15	ID
16	InheritanceAttribute
17	Inherited
18	IsDirty
19	ReadOnly
20	ShadowProperties
21	ShouldCodeSerialize
22	Verbs
23	Methods:
24	GetDesignTimeHtml
25	

1	
2	[C#] public override string GetDesignTimeHtml();
3	[C++] public: String* GetDesignTimeHtml();
4	[VB] Overrides Public Function GetDesignTimeHtml() As String
5	[JScript] public override function GetDesignTimeHtml(): String;
6	
7	Description
8	Gets the HTML that is used to represent the control at design time.
9	Return Value: The HTML that is used to represent the control at design time.
10	BaseDataListComponentEditor class
11	(System.Web.UI.Design.WebControls)
12	UpdateDesignTimeHtml
13	
14	
15	Description
16	Provides a base component editor class for the Web Forms
17	System.Web.UI.WebControls.DataGrid and
18	System.Web.UI.WebControls.DataList Web server controls.
19	BaseDataListComponentEditor
20	Example Syntax:
21	UpdateDesignTimeHtml
22	
23	[C#] public BaseDataListComponentEditor(int initialPage);
24	[C++] public: BaseDataListComponentEditor(int initialPage);
25	[VB] Public Sub New(ByVal initialPage As Integer)

[JScript] public function BaseDataListComponentEditor(initialPage: int); 2 Description 3 Initializes a new instance of the System.Web.UI.Design.WebControls.BaseDataListComponentEditor class. 5 The index of the initial page to display. 6 EditComponent 7 8 [C#] public override bool EditComponent(ITypeDescriptorContext context, object 9 obj, IWin32Window parent); 10 [C++] public: bool EditComponent(ITypeDescriptorContext\* context, Object\* 11 obj, IWin32Window\* parent); 12 [VB] Overrides Public Function EditComponent(ByVal context As 13 ITypeDescriptorContext, ByVal obj As Object, ByVal parent As IWin32Window) 14 As Boolean 15 [JScript] public override function EditComponent(context: 16 ITypeDescriptorContext, obj : Object, parent : IWin32Window) : Boolean; 17 18 Description 19 Edits the specified component using the specified context descriptor and 20 parent window. 21 Return Value: true if editing the component succeeded; otherwise, false. An 22 System.ComponentModel.ITypeDescriptorContext that can be used to gain 23 additional context information. The component to edit. The parent window. 24 GetInitialComponentEditorPageIndex 25

[C#] protected override int GetInitialComponentEditorPageIndex();
[C++] protected: int GetInitialComponentEditorPageIndex();
[VB] Overrides Protected Function GetInitialComponentEditorPageIndex() As
Integer
[JScript] protected override function GetInitialComponentEditorPageIndex(): int;
Description
Gets the index of the initial component editor page.
Return Value: The index of the initial page.
BaseDataListDesigner class (System.Web.UI.Design.WebControls)
ToString
Description
Provides a base designer class for the
System.Web.UI.WebControls.DataList Web server control.
BaseDataListDesigner
Example Syntax:
ToString
[C#] public BaseDataListDesigner();
[C++] public: BaseDataListDesigner();
[VB] Public Sub New()
[JScript] public function BaseDataListDesigner();

1 Description 2 Initializes a new instance of the 3  $System. Web. UI. Design. WebControls. Base Data List Designer \ class.$ ActiveTemplateEditingFrame 5 AllowResize 6 AssociatedComponents 7 Behavior 8 Can Enter Template Mode9 Component 10 **DataBindings** 11 DataKeyField 12 **ToString** 13 14 15 Description 16 Gets or sets the value of the control's data key field. 17 The 18 System. Web. UI. Design. WebControls. BaseDataListDesigner. Data KeyField19 property of this designer gets or sets the 20  ${\bf System. Web. UI. WebControls. Data List. Data Key Field}\ property\ of\ the$ 21 System.Web.UI.WebControls.DataList control that this designer is editing. 22 DataMember 23 **ToString** 24 25

[C#] public string DataMember {get; set;}
[C++] public:property String* get_DataMember();public:property void
set_DataMember(String*);
[VB] Public Property DataMember As String
[JScript] public function get DataMember(): String; public function set
DataMember(String);
Description
Gets or sets the value of the control's data member field.
The
System.Web.UI.Design.WebControls.BaseDataListDesigner.DataMember
property of this designer gets or sets the
System.Web.UI.WebControls.DataList.DataMember property of the
System.Web.UI.WebControls.DataList control that this designer is editing.
DataSource
ToString
[C#] public string DataSource {get; set;}
[C++] public:property String* get_DataSource();public:property void
set_DataSource(String*);
[VB] Public Property DataSource As String
[JScript] public function get DataSource(): String; public function set
DataSource(String);

20

21

22

23

24

25

-		. •
Desc	rın	tini
$\mathcal{L}_{\mathcal{L}_{\mathcal{S}}\mathcal{L}_{\mathcal{S}}}$	$\iota \iota \nu$	uvo.

1

2

Gets or sets the value of the control's data source property.

The

System.Web.UI.Design.WebControls.BaseDataListDesigner.DataSource property of this designer gets or sets the

System.Web.UI.WebControls.DataList.DataSource property of the

System.Web.UI.WebControls.DataList control that this designer is editing.

DesignTimeElement

DesignTimeElementView

DesignTimeHtmlRequiresLoadComplete

**ToString** 

### Description

Indicates whether loading must have completed to display the design-time HTML.

If this property is set to **true**, the control will not be displayed until the entire load has completed.

Hide Properties In Template Mode

ID

InheritanceAttribute

Inherited

InTemplateMode

**IsDirty** 

	1	ReadOnly
	2	ShadowProper
	3	ShouldCodeSe
	4	Verbs
	5	ToString
	6	
	7	
	8	Description
	9	Gets or sets th
	10	Dispose
	11	
	12	[C#] protected overr
	13	[C++] protected: voi
14	14	[VB] Overrides Prot
	15	[JScript] protected o
	16	
in the	17	Description
	18	Releases the
	19	System.Web.UI.De
	20	releases the manage
	21	This method
	22	System.Object.Fin
	23	resources; false to re
	24	GetDesignTi
	25	

rties erialize

he collection of verbs available to this designer.

ride void Dispose(bool disposing);

id Dispose(bool disposing);

tected Sub Dispose(ByVal disposing As Boolean)

override function Dispose(disposing : Boolean);

unmanaged resources used by the

esign.WebControls.BaseDataListDesigner and optionally ed resources.

is called by the public Dispose() method and the alize method. true to release both managed and unmanaged elease only unmanaged resources.

meDataSource

[C#] protected IEnumerable GetDesignTimeDataSource(int minimumRows, out bool dummyDataSource);

[C++] protected: IEnumerable\* GetDesignTimeDataSource(int minimumRows, bool\* dummyDataSource);

[VB] Protected Function GetDesignTimeDataSource(ByVal minimumRows As Integer, ByRef dummyDataSource As Boolean) As IEnumerable
[JScript] protected function GetDesignTimeDataSource(minimumRows: int, dummyDataSource: Boolean): IEnumerable; Gets a set of sample data that matches the schema of the selected data source.

#### Description

Gets a set of sample data that matching the schema of the selected data source.

Return Value: A live data source for use at design-time. The minimum rows of sample data that the data source data should contain. A boolean variable that will be set to **true** if the returned data source contains dummy data, or **false** if the data source was populated with data from an actual data source.

### GetDesignTimeDataSource

[C#] protected IEnumerable GetDesignTimeDataSource(IEnumerable selectedDataSource, int minimumRows, out bool dummyDataSource);
[C++] protected: IEnumerable\* GetDesignTimeDataSource(IEnumerable\* selectedDataSource, int minimumRows, bool\* dummyDataSource);
[VB] Protected Function GetDesignTimeDataSource(ByVal selectedDataSource

3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	

19

20

21

22

23

24

As IEnumerable, ByVal minimumRows As Integer, ByRef dummyDataSource As Boolean) As IEnumerable

[JScript] protected function GetDesignTimeDataSource(selectedDataSource:

IEnumerable, minimumRows: int, dummyDataSource: Boolean): IEnumerable;

## Description

Gets a set of sample data that matches the schema of the selected data source.

Return Value: A live data source for use at design-time. The data source that will be used as a template for the format of the data. The minimum number of rows of sample data that the data source data should contain. A boolean variable that will be set to **true** if the returned data source contains dummy data, or **false** if the data source was populated with data from an actual data source.

GetResolvedSelectedDataSource

[C#] public IEnumerable GetResolvedSelectedDataSource();

[C++] public: sealed IEnumerable\* GetResolvedSelectedDataSource();

[VB] NotOverridable Public Function GetResolvedSelectedDataSource() As

**IEnumerable** 

[JScript] public function GetResolvedSelectedDataSource(): IEnumerable;

#### Description

Gets the data member that is currently selected within the data source currently bound to the control.

Return Value: The currently selected data member, or null if the control was not

bound to a data source, or the data source, site of the designer's component, or the container of the data source could not be accessed.

**GetSelectedDataSource** 

[C#] public object GetSelectedDataSource();

[C++] public: \_\_sealed Object\* GetSelectedDataSource();

[VB] NotOverridable Public Function GetSelectedDataSource() As Object

[JScript] public function GetSelectedDataSource(): Object;

### Description

Gets the selected data source component from the component's container.

Return Value: The selected data source, or null if a data source is not found, or if a data source with the same name does not exist.

GetTemplateContainerDataSource

[C#] public override IEnumerable GetTemplateContainerDataSource(string templateName);

[C++] public: IEnumerable\* GetTemplateContainerDataSource(String\* templateName);

[VB] Overrides Public Function GetTemplateContainerDataSource(ByVal templateName As String) As IEnumerable

[JScript] public override function

GetTemplateContainerDataSource(templateName: String): IEnumerable;

Description

22

23

24

2

3

4

5

6

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets the template's container's data source.

Return Value: An object that implements **System.Collections.IEnumerable** and contains the data source or data sources available to the template's container. The name of the template to retrieve the data source for.

Initialize

[C#] public override void Initialize(IComponent component);

[C++] public: void Initialize(IComponent\* component);

[VB] Overrides Public Sub Initialize(ByVal component As IComponent)

[JScript] public override function Initialize(component : IComponent);

Description

Initializes the designer with the **System.Web.UI.WebControls.DataGrid** control that this instance of the designer is associated with. The associated **System.Web.UI.WebControls.DataGrid** control.

InvokePropertyBuilder

[C#] protected internal void InvokePropertyBuilder(int initialPage);

[C++] protected public: void InvokePropertyBuilder(int initialPage);

[VB] Protected Friend Dim Sub InvokePropertyBuilder(ByVal initialPage As Integer)

[JScript] package function InvokePropertyBuilder(initialPage: int);

Description

25

3

5

6

7

8

9

10

11

12

13

14

15

17

18

19

20

21

22

23

24

25

Invokes the property builder beginning with the specified page. The page to begin with.

**OnAutoFormat** 

[C#] protected void OnAutoFormat(object sender, EventArgs e);

[C++] protected: void OnAutoFormat(Object\* sender, EventArgs\* e);

[VB] Protected Sub OnAutoFormat(ByVal sender As Object, ByVal e As

EventArgs)

[JScript] protected function OnAutoFormat(sender : Object, e : EventArgs);

Description

Represents the method that will handle the AutoFormat event. The source of the event. An **System.EventArgs** object that provides data about the event.

**OnComponentChanged** 

[C#] public override void OnComponentChanged(object sender,

ComponentChangedEventArgs e);

[C++] public: void OnComponentChanged(Object\* sender,

ComponentChangedEventArgs\* e);

[VB] Overrides Public Sub OnComponentChanged(ByVal sender As Object,

ByVal e As ComponentChangedEventArgs)

[JScript] public override function OnComponentChanged(sender : Object, e :

ComponentChangedEventArgs);

Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Represents the method that will handle the component change event. The source of the event. The

System.ComponentModel.Design.ComponentChangedEventArgs that provides data about the event.

OnDataSourceChanged

[C#] protected internal virtual void OnDataSourceChanged();

[C++] protected public: virtual void OnDataSourceChanged();

[VB] Overridable Protected Friend Dim Sub OnDataSourceChanged()

[JScript] package function OnDataSourceChanged();

## Description

Raises the DataSourceChanged event.

OnPropertyBuilder

[C#] protected void OnPropertyBuilder(object sender, EventArgs e);

[C++] protected: void OnPropertyBuilder(Object\* sender, EventArgs\* e);

[VB] Protected Sub OnPropertyBuilder(ByVal sender As Object, ByVal e As

EventArgs)

[JScript] protected function OnPropertyBuilder(sender : Object, e : EventArgs);

## Description

Represents the method that will handle the property builder event. The source of the event. An **System.EventArgs** object that provides data about the event.

3

5

6

7

8

9

10

11

12

13

14

15

17

18

19

20

21

22

23

25

OnSty	/lesChanged
Ollon	/168Changeu

[C#] protected internal void OnStylesChanged();

[C++] protected public: void OnStylesChanged();

[VB] Protected Friend Dim Sub OnStylesChanged()

[JScript] package function OnStylesChanged();

## Description

Provides a method that can be overrriden to implement functionality that should occur when a style of the control has been changed.

This method can serve as notification that a style of the control has changed.

OnTemplateEditingVerbsChanged

[C#] protected abstract void OnTemplateEditingVerbsChanged();

[C++] protected: virtual void OnTemplateEditingVerbsChanged() = 0;

[VB] MustOverride Protected Sub OnTemplateEditingVerbsChanged()

[JScript] protected abstract function OnTemplateEditingVerbsChanged();

# Description

Provides a method that can be overrriden to implement functionality that should occur when the designer's template editing verbs have changed.

This method can serve as notification that the designer's template editing verbs have changed.

PreFilterProperties

П	
1	
2	[C#] protected override void PreFilterProperties(IDictionary properties);
3	[C++] protected: void PreFilterProperties(IDictionary* properties);
4	[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As
5	IDictionary)
6	[JScript] protected override function PreFilterProperties(properties : IDictionary);
7	
8	Description
9	Filters the properties exposed through a
10	System.ComponentModel.TypeDescriptor and replaces the runtime DataSource
11	property descriptor with a design-time data source. The set of properties to filter.
12	BaseValidatorDesigner class (System.Web.UI.Design.WebControls)
13	UpdateDesignTimeHtml
14	
15	
16	Description
17	Provides design-time support for controls that derive from
18	System.Web.UI.WebControls.BaseValidator.
19	BaseValidatorDesigner
20	Example Syntax:
21	UpdateDesignTimeHtml
22	
23	[C#] public BaseValidatorDesigner();
24	[C++] public: BaseValidatorDesigner();
25	

	1	[VB] Public Sub New()
	2	[JScript] public function BaseValidatorDesigner();
	3	AllowResize
	4	AssociatedComponents
	5	Behavior
	6	Component
	7	DataBindings
	8	DesignTimeElement
=	9	DesignTimeElementView
	10	DesignTimeHtmlRequiresLoadComplete
	11	ID
	12	InheritanceAttribute
	13	Inherited
Red Tark Nº Bros Sank Spot	14	IsDirty
	15	ReadOnly
	16	ShadowProperties
:3 :14	17	ShouldCodeSerialize
	18	Verbs
	19	GetDesignTimeHtml
	20	
	21	[C#] public override string GetDesignTimeHtml();
	22	[C++] public: String* GetDesignTimeHtml();
	23	[VB] Overrides Public Function GetDesignTimeHtml() As String
	24	[JScript] public override function GetDesignTimeHtml() : String;
	25	

	11	
	1	
	2	Description
	3	Gets the HTML that is used to represent the control at design time.
	4	Return Value: The HTML that is used to represent the control at design time.
	5	ButtonDesigner class (System.Web.UI.Design.WebControls)
	6	UpdateDesignTimeHtml
	7	
	8	
	9	Description
mii	10	Provides design-time support for the
	11	System.Web.UI.WebControls.Button Web server control.
	12	ButtonDesigner
	13	Example Syntax:
	14	UpdateDesignTimeHtml
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	15	
to the three to the first	16	[C#] public ButtonDesigner();
	17	[C++] public: ButtonDesigner();
	18	[VB] Public Sub New()
	19	[JScript] public function ButtonDesigner();
	20	AllowResize
	21	AssociatedComponents
	22	Behavior
	23	Component
	24	DataBindings
	25	DesignTimeElement
		•

1	DesignTimeElementView
2	DesignTimeHtmlRequiresLoadComplete
3	ID
4	InheritanceAttribute
5	Inherited
6	IsDirty
7	ReadOnly
8	ShadowProperties
9	ShouldCodeSerialize
10	Verbs
11	GetDesignTimeHtml
12	
13	[C#] public override string GetDesignTimeHtml();
14	[C++] public: String* GetDesignTimeHtml();
15	[VB] Overrides Public Function GetDesignTimeHtml() As String
16	[JScript] public override function GetDesignTimeHtml(): String;
17	
18	Description
19	Gets the HTML that is used to represent the control at design time.
20	Return Value: The HTML that is used to represent the control at design time.
21	CalendarAutoFormatDialog class (System.Web.UI.Design.WebControls
22	UpdateDesignTimeHtml
23	
24	
25	Description

1	Provides an AutoFormat page for a
2	System.Web.UI.WebControls.Calendar Web server control.
3	CalendarAutoFormatDialog
4	Example Syntax:
5	UpdateDesignTimeHtml
6	
7	[C#] public CalendarAutoFormatDialog(Calendar calendar);
8	[C++] public: CalendarAutoFormatDialog(Calendar* calendar);
9	[VB] Public Sub New(ByVal calendar As Calendar)
10	[JScript] public function CalendarAutoFormatDialog(calendar : Calendar);
11	
12	Description
13	Create a new AutoFormatPage instance Create a new AutoFormatPage
14	instance
15	AcceptButton
16	AccessibilityObject
17	AccessibleDefaultActionDescription
18	AccessibleDescription
19	AccessibleName
20	AccessibleRole
21	ActiveControl
22	ActiveMdiChild
23	AllowDrop
24	AllowTransparency
25	Anchor

	1	AutoScale
	2	AutoScaleBaseSize
	3	AutoScroll
	4	AutoScrollMargin
	5	AutoScrollMinSize
	6	AutoScrollPosition
	7	BackColor
	8	BackgroundImage
	9	BindingContext
=======================================	10	Bottom
	11	Bounds
assi tini and the this that the the	12	CancelButton
And had	13	CanFocus
	14	CanSelect
ii" tuni tani ii dam tuni	15	Capture
y yang y	16	CausesValidation
	17	ClientRectangle
	18	ClientSize
	19	CompanyName
	20	Container
	21	ContainsFocus
	22	ContextMenu
	23	ControlBox
	24	Controls
	25	Created

	1	CreateParams
	2	Cursor
	3	DataBindings
	4	DefaultImeMode
	5	DefaultSize
	6	DesignMode
	7	DesktopBounds
	8	DesktopLocation
	9	DialogResult
)	10	DisplayRectangle
	11	Disposing
	12	Dock
	13	DockPadding
	14	Enabled
	15	Events
	16	Focused
	17	Font
	18	FontHeight
	19	ForeColor
	20	FormBorderStyle
	21	Handle
	22	HasChildren
	23	Height
	24	HelpButton
	25	HScroll

1	Icon
2	ImeMode
3	InvokeRequired
4	IsAccessible
5	IsDisposed
6	IsHandleCreated
7	IsMdiChild
8	IsMdiContainer
9	IsRestrictedWindow
10	KeyPreview
11	Left
12	Location
13	MaximizeBox
14	MaximizedBounds
15	MaximumSize
16	MdiChildren
17	MdiParent
18	Menu
19	MergedMenu
20	MinimizeBox
21	MinimumSize
22	Modal
23	Name
24	Opacity
25	OwnedForms

1	Owner
2	Parent
3	ParentForm
4	ProductName
5	ProductVersion
6	RecreatingHandle
7	Region
8	RenderRightToLeft
9	ResizeRedraw
10	Right
11	RightToLeft
12	ShowFocusCues
13	ShowInTaskbar
14	ShowKeyboardCues
15	Site
16	Size
17	SizeGripStyle
18	StartPosition
19	TabIndex
20	TabStop
21	Tag
22	Text
23	Тор
24	TopLevel
25	TopLevelControl

1	TopMost
2	TransparencyKey
3	Visible
4	VScroll
5	Width
6	WindowState
7	WindowTarget
8	DoDelayLoadActions
9	
10	[C#] protected void DoDelayLoadActions();
11	[C++] protected: void DoDelayLoadActions();
12	[VB] Protected Sub DoDelayLoadActions()
13	[JScript] protected function DoDelayLoadActions();
14	
15	Description
16	Executes any initialization that was delayed until the first idle time
17	Executes any initialization that was delayed until the first idle time
18	OnActivated
19	
20	[C#] protected void OnActivated(object source, EventArgs e);
21	[C++] protected: void OnActivated(Object* source, EventArgs* e);
22	[VB] Protected Sub OnActivated(ByVal source As Object, ByVal e As
23	EventArgs)
24	[JScript] protected function OnActivated(source : Object, e : EventArgs);
25	

Description

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

25

Handles the activated event of the

System.Web.UI.WebControls.Calendar AutoFormat dialog.

This method provides an opportunity to perform operations after a Calendar AutoFormat dialog has been activated. The source of the event. An **System.EventArgs** that provides information about the event.

OnOKClicked

[C#] protected void OnOKClicked(object source, EventArgs e);

[C++] protected: void OnOKClicked(Object\* source, EventArgs\* e);

[VB] Protected Sub OnOKClicked(ByVal source As Object, ByVal e As

EventArgs)

[JScript] protected function OnOKClicked(source : Object, e : EventArgs);

Description

Handle changes in the pre-defined schema choices Handle changes in the pre-defined schema choices

OnSelChangedScheme

[C#] protected void OnSelChangedScheme(object source, EventArgs e);

[C++] protected: void OnSelChangedScheme(Object\* source, EventArgs\* e);

[VB] Protected Sub OnSelChangedScheme(ByVal source As Object, ByVal e As

EventArgs)

[JScript] protected function OnSelChangedScheme(source : Object, e :

- 11	
1	EventArgs);
2	
3	Description
4	Handle changes in the pre-defined schema choices Handle changes in the
5	pre-defined schema choices
6	SaveComponent
7	
8	[C#] protected void SaveComponent();
9	[C++] protected: void SaveComponent();
10	[VB] Protected Sub SaveComponent()
11	[JScript] protected function SaveComponent();
12	
13	Description
14	Save any changes into the component Save any changes into the component
15	CalendarDesigner class (System.Web.UI.Design.WebControls)
16	WndProc
17	
18	
19	Description
20	Provides design-time support for the
21	System.Web.UI.WebControls.Calendar Web server control.
22	CalendarDesigner
23	Example Syntax:
24	WndProc
25	
	••

	1	
	2	[C#] public CalendarDesigner();
	3	[C++] public: CalendarDesigner();
	4	[VB] Public Sub New()
	5	[JScript] public function CalendarDesigner();
	6	AllowResize
	7	AssociatedComponents
	8	Behavior
	9	Component
	10	DataBindings
	11	DesignTimeElement
	12	DesignTimeElementView
	13	DesignTimeHtmlRequiresLoadComplete
	14	ID
the trust that the trust that	15	InheritanceAttribute
	16	Inherited
***************************************	17	IsDirty
	18	ReadOnly
	19	ShadowProperties
	20	ShouldCodeSerialize
	21	Verbs
	22	WndProc
	23	
	24	
	25	Description

1	Gets the set of verbs available to this designer.
2	Initialize
3	
4	[C#] public override void Initialize(IComponent component);
5	[C++] public: void Initialize(IComponent* component);
6	[VB] Overrides Public Sub Initialize(ByVal component As IComponent)
7	[JScript] public override function Initialize(component : IComponent);
8	
9	Description
10	Initializes the designer with the component for design.
11	The designer host calls
12	System. Web. UI. Design. WebControls. Calendar Designer. Initialize (System. Controls) and the property of t
13	mponentModel.IComponent) to establish the component for design. The control
14	element for design.
15	OnAutoFormat
16	
17	[C#] protected void OnAutoFormat(object sender, EventArgs e);
18	[C++] protected: void OnAutoFormat(Object* sender, EventArgs* e);
19	[VB] Protected Sub OnAutoFormat(ByVal sender As Object, ByVal e As
20	EventArgs)
21	[JScript] protected function OnAutoFormat(sender : Object, e : EventArgs);
22	
23	Description
24	Delegate to handle the AutoFormat verb that displays a new
25	System.Web.UI.Design.WebControls.CalendarAutoFormatDialog . The

1	source of the event. The System. EventArgs object that provides data about the
2	event.
3	CheckBoxDesigner class (System.Web.UI.Design.WebControls)
4	UpdateDesignTimeHtml
5	
6	
7	Description
8	Provides design-time support for the
9	System.Web.UI.WebControls.CheckBox Web server control.
10	CheckBoxDesigner
11	Example Syntax:
12	UpdateDesignTimeHtml
13	
14	[C#] public CheckBoxDesigner();
15	[C++] public: CheckBoxDesigner();
16	[VB] Public Sub New()
17	[JScript] public function CheckBoxDesigner();
18	AllowResize
19	AssociatedComponents
20	Behavior
21	Component
22	DataBindings
23	DesignTimeElement
24	DesignTimeElementView
25	DesignTimeHtmlRequiresLoadComplete

1	ID
2	InheritanceAttribute
3	Inherited
4	IsDirty
5	ReadOnly
6	ShadowProperties
7	ShouldCodeSerialize
8	Verbs
9	GetDesignTimeHtml
10	
11	[C#] public override string GetDesignTimeHtml();
12	[C++] public: String* GetDesignTimeHtml();
13	[VB] Overrides Public Function GetDesignTimeHtml() As String
14	[JScript] public override function GetDesignTimeHtml(): String;
15	
16	Description
17	Gets the HTML that is used to represent the control at design time.
18	Return Value: The HTML that is used to represent the control at design time.
19	DataGridColumnCollectionEditor class
20	(System.Web.UI.Design.WebControls)
21	UpdateDesignTimeHtml
22	
23	
24	Description
25	

	1	Provides a collection editing user interface for a column collection of a
	2	System.Web.UI.WebControls.DataGrid Web server control.
	3	DataGridColumnCollectionEditor
	4	Example Syntax:
	5	UpdateDesignTimeHtml
	6	
	7	[C#] public DataGridColumnCollectionEditor();
	8	[C++] public: DataGridColumnCollectionEditor();
	9	[VB] Public Sub New()
	10	[JScript] public function DataGridColumnCollectionEditor();
	11	EditValue
	12	
	13	[C#] public override object EditValue(ITypeDescriptorContext context,
	14	IServiceProvider provider, object value);
	15	[C++] public: Object* EditValue(ITypeDescriptorContext* context,
	16	IServiceProvider* provider, Object* value);
is	17	[VB] Overrides Public Function EditValue(ByVal context As
	18	ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As
	19	Object) As Object
	20	[JScript] public override function EditValue(context : ITypeDescriptorContext,
	21	provider : IServiceProvider, value : Object) : Object;
	22	
	23	Description
	24	Edits the specified value using the specified context descriptor and service
	25	provider.

1	Return Value: The new value. An
2	System.ComponentModel.ITypeDescriptorContext that indicates the context of
3	the object to edit the value of. An System.IServiceProvider. The object to edit.
4	GetEditStyle
5	
6	[C#] public override UITypeEditorEditStyle
7	GetEditStyle(ITypeDescriptorContext context);
8	[C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext*
9	context);
10	[VB] Overrides Public Function GetEditStyle(ByVal context As
11	ITypeDescriptorContext) As UITypeEditorEditStyle
12	[JScript] public override function GetEditStyle(context : ITypeDescriptorContext)
13	: UITypeEditorEditStyle;
14	
15	Description
16	Gets the edit style used by the
17	System.Web.UI.Design.WebControls.DataGridColumnCollectionEditor.Edit
18	Value(System.ComponentModel.ITypeDescriptorContext,System.IServicePro
19	vider, System. Object) method.
20	Return Value: A System.Drawing.Design.UITypeEditorEditStyle that
21	represents the editor style that this editor uses. An
22	System.ComponentModel.ITypeDescriptorContext that indicates the context of
23	the object to edit the value of.
24	DataGridComponentEditor class (System.Web.UI.Design.WebControls)
25	ToString

	1	
	2	
	3	Description
	4	Provides a component editor for a Web Forms
	5	System.Web.UI.WebControls.DataGrid Web server control.
	6	DataGridComponentEditor
	7	Example Syntax:
	8	ToString
	9	
	10	[C#] public DataGridComponentEditor();
	11	[C++] public: DataGridComponentEditor();
	12	[VB] Public Sub New()
	13	[JScript] public function DataGridComponentEditor(); Initializes a new instance
	14	of the System. Web. UI. Design. WebControls. Data Grid Component Editor class.
	15	
1	16	Description
dia.	17	Initializes a new instance of the
	18	System.Web.UI.Design.WebControls.DataGridComponentEditor class.
	19	DataGridComponentEditor
	20	Example Syntax:
	21	ToString
	22	
	23	[C#] public DataGridComponentEditor(int initialPage);
	24	[C++] public: DataGridComponentEditor(int initialPage);
	25	[VB] Public Sub New(ByVal initialPage As Integer)

1	[JScript] public function DataGridComponentEditor(initialPage : int);
2	
3	Description
4	Initializes a new instance of the
5	System.Web.UI.Design.WebControls.DataGridComponentEditor class. The
6	index of the initial page.
7	GetComponentEditorPages
8	
9	[C#] protected override Type[] GetComponentEditorPages();
10	[C++] protected: Type* GetComponentEditorPages() [];
11	[VB] Overrides Protected Function GetComponentEditorPages() As Type()
12	[JScript] protected override function GetComponentEditorPages(): Type[];
13	
14	Description
15	Gets the set of all pages in the System.Web.UI.WebControls.DataGrid
16	control.
17	Return Value: An array consisting of the set of component editor pages.
18	This method can be overridden to change the set of pages to show.
19	DataGridDesigner class (System.Web.UI.Design.WebControls)
20	ToString
21	
22	
23	Description
24	Provides design-time support for the
25	System.Web.UI.WebControls.DataGrid Web server control.

1	DataGridDesigner
2	Example Syntax:
3	ToString
4	
5	[C#] public DataGridDesigner();
6	[C++] public: DataGridDesigner();
7	[VB] Public Sub New()
8	[JScript] public function DataGridDesigner();
9	
10	Description
11	Initializes a new instance of the
12	System.Web.UI.Design.WebControls.DataGridDesigner class.
13	ActiveTemplateEditingFrame
14	AllowResize
15	AssociatedComponents
16	Behavior
17	CanEnterTemplateMode
18	Component
19	DataBindings
20	DataKeyField
21	DataMember
22	DataSource
23	DesignTimeElement
24	DesignTimeElementView
25	DesignTimeHtmlRequiresLoadComplete

1	HidePropertiesInTemplateMode
2	ID
3	InheritanceAttribute
4	Inherited
5	InTemplateMode
6	IsDirty
7	ReadOnly
8	ShadowProperties
9	ShouldCodeSerialize
10	Verbs
11	CreateTemplateEditingFrame
12	
13	[C#] protected override ITemplateEditingFrame
14	CreateTemplateEditingFrame(TemplateEditingVerb verb);
15	[C++] protected: ITemplateEditingFrame*
16	CreateTemplateEditingFrame(TemplateEditingVerb* verb);
17	[VB] Overrides Protected Function CreateTemplateEditingFrame(ByVal verb As
18	TemplateEditingVerb) As ITemplateEditingFrame
19	[JScript] protected override function CreateTemplateEditingFrame(verb:
20	TemplateEditingVerb): ITemplateEditingFrame;
21	
22	Description
23	Creates a template editing frame using the specified verb.
24	Return Value: The new template editing frame. The verb to create the template
25	editing frame for.

Description

Dispose

2	
3	[C#] protected override void Dispose(bool disposing);
4	[C++] protected: void Dispose(bool disposing);
5	[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)
6	[JScript] protected override function Dispose(disposing : Boolean);
7	
8	Description
9	Releases the unmanaged resources used by the
10	System.Web.UI.Design.WebControls.DataGridDesigner and optionally
11	releases the managed resources.
12	This method is called by the public <b>Dispose()</b> method and the
13	System.Object.Finalize method. true to release both managed and unmanaged
14	resources; false to release only unmanaged resources.
15	GetCachedTemplateEditingVerbs
16	
17	[C#] protected override TemplateEditingVerb[]
18	GetCachedTemplateEditingVerbs();
19	[C++] protected: TemplateEditingVerb* GetCachedTemplateEditingVerbs() [];
20	[VB] Overrides Protected Function GetCachedTemplateEditingVerbs() As
21	TemplateEditingVerb()
22	[JScript] protected override function GetCachedTemplateEditingVerbs():
23	TemplateEditingVerb[];

1	Gets the cached template editing verbs.
2	Return Value: An array containing the cached template editing verbs.
3	GetDesignTimeHtml
4	
5	[C#] public override string GetDesignTimeHtml();
6	[C++] public: String* GetDesignTimeHtml();
7	[VB] Overrides Public Function GetDesignTimeHtml() As String
8	[JScript] public override function GetDesignTimeHtml(): String;
9	
10	Description
11	Gets the HTML that is used to represent the control at design time.
12	Return Value: The HTML that is used to represent the control at design time.
13	GetEmptyDesignTimeHtml
14	
15	[C#] protected override string GetEmptyDesignTimeHtml();
16	[C++] protected: String* GetEmptyDesignTimeHtml();
17	[VB] Overrides Protected Function GetEmptyDesignTimeHtml() As String
18	[JScript] protected override function GetEmptyDesignTimeHtml(): String;
19	
20	Description
21	Gets the HTML used to represent an empty template-based control at
22	design time.
23	Return Value: The HTML used to represent an empty template-based control at
24	design time.
25	GetErrorDesignTimeHtml

18

19

20

21

22

23

24

[C++] protected: String\* GetErrorDesignTimeHtml(Exception\* e); [VB] Overrides Protected Function GetErrorDesignTimeHtml(ByVal e As Exception) As String [JScript] protected override function GetErrorDesignTimeHtml(e: Exception): String; 8 Description Gets the HTML displayed at design-time for the specified exception when 10 an error has been encountered while rendering the control. 11 Return Value: The HTML displayed at design-time for the specified exception. 12 The exception to display an error message for. 13 GetTemplateContainerDataItemProperty 14 15 16

[C#] protected override string GetErrorDesignTimeHtml(Exception e);

[C#] public override string GetTemplateContainerDataItemProperty(string templateName);

[C++] public: String\* GetTemplateContainerDataItemProperty(String\* templateName);

[VB] Overrides Public Function GetTemplateContainerDataItemProperty(ByVal templateName As String) As String

[JScript] public override function

GetTemplateContainerDataItemProperty(templateName: String): String;

Description

21

22

23

Gets the template's container's data item property.

Return Value: The template's container's data item property. The name of the template.

#### GetTemplateContent

[C#] public override string GetTemplateContent(ITemplateEditingFrame editingFrame, string templateName, out bool allowEditing);

[C++] public: String\* GetTemplateContent(ITemplateEditingFrame\* editingFrame, String\* templateName, bool\* allowEditing);

[VB] Overrides Public Function GetTemplateContent(ByVal editingFrame As ITemplateEditingFrame, ByVal templateName As String, ByRef allowEditing As Boolean) As String

[JScript] public override function GetTemplateContent(editingFrame :

ITemplateEditingFrame, templateName: String, allowEditing: Boolean): String;

## Description

Gets the template's content.

Return Value: The template's content. The template frame to retrieve content for.

The name of the template. A boolean variable that will be set to **true** if the

template's content can be edited, or false if the content is read-only.

GetTemplatePropertyParentType

[C#] public override Type GetTemplatePropertyParentType(string templateName);

[C++] public: Type\* GetTemplatePropertyParentType(String\* templateName);

1	[VB] Overrides Public Function GetTemplatePropertyParentType(ByVal
2	templateName As String) As Type
3	[JScript] public override function GetTemplatePropertyParentType(templateName
4	: String) : Type;
5	
6	Description
7	Gets the type of the parent of the template property.
8	Return Value: The type of the object that has the template property. The name of
9	the template to return the type of the parent for.
10	Initialize
11	
12	[C#] public override void Initialize(IComponent component);
13	[C++] public: void Initialize(IComponent* component);
14	[VB] Overrides Public Sub Initialize(ByVal component As IComponent)
15	[JScript] public override function Initialize(component : IComponent);
16	
17	Description
18	Initializes the designer with the System.Web.UI.WebControls.DataGrid
19	control that this instance of the designer is associated with. The associated
20	System.Web.UI.WebControls.DataGrid control.
21	OnColumnsChanged
22	
23	[C#] public virtual void OnColumnsChanged();
24	[C++] public: virtual void OnColumnsChanged();
25	[VB] Overridable Public Sub OnColumnsChanged()

[JScript] public function OnColumnsChanged(); 2 Description Notification that is called when the columns changed event occurs. OnTemplateEditingVerbsChanged 5 6 [C#] protected override void OnTemplateEditingVerbsChanged(); [C++] protected: void OnTemplateEditingVerbsChanged(); 8 [VB] Overrides Protected Sub OnTemplateEditingVerbsChanged() 9 [JScript] protected override function OnTemplateEditingVerbsChanged(); 10 11 Description 12 Provides an opportunity to do processing or other actions when a change 13 has been made to the template editing verbs. 14 SetTemplateContent 15 16 [C#] public override void SetTemplateContent(ITemplateEditingFrame 17 editingFrame, string templateName, string templateContent); 18 [C++] public: void SetTemplateContent(ITemplateEditingFrame\* editingFrame, 19 String\* templateName, String\* templateContent); 20 [VB] Overrides Public Sub SetTemplateContent(ByVal editingFrame As 21 ITemplateEditingFrame, ByVal templateName As String, ByVal templateContent 22 As String) 23 [JScript] public override function SetTemplateContent(editingFrame: ITemplateEditingFrame, templateName: String, templateContent: String);

- 1	
1	
2	Description
3	Sets the content for the specified template and frame. The template frame to
4	set the content for. The name of the template. The content to set.
5	DataListComponentEditor class (System.Web.UI.Design.WebControls)
6	UpdateDesignTimeHtml
7	
8	
9	Description
10	Provides a component editor for a Web Forms
11	System.Web.UI.WebControls.DataList control.
12	DataListComponentEditor
13	Example Syntax:
14	UpdateDesignTimeHtml
15	
16	[C#] public DataListComponentEditor();
17	[C++] public: DataListComponentEditor();
18	[VB] Public Sub New()
19	[JScript] public function DataListComponentEditor(); Initializes a new instance of
20	System.Web.UI.Design.WebControls.DataListComponentEditor.
21	
22	Description
23	Initializes a new instance of
24	System.Web.UI.Design.WebControls.DataListComponentEditor .
25	DataListComponentEditor

1	Example Syntax:
2	UpdateDesignTimeHtml
3	
4	[C#] public DataListComponentEditor(int initialPage);
5	[C++] public: DataListComponentEditor(int initialPage);
6	[VB] Public Sub New(ByVal initialPage As Integer)
7	[JScript] public function DataListComponentEditor(initialPage : int);
8	
9	Description
10	Initializes a new instance of
11	System.Web.UI.Design.WebControls.DataListComponentEditor . The index
12	of the initial page to display.
13	GetComponentEditorPages
14	
15	[C#] protected override Type[] GetComponentEditorPages();
16	[C++] protected: Type* GetComponentEditorPages() [];
17	[VB] Overrides Protected Function GetComponentEditorPages() As Type()
18	[JScript] protected override function GetComponentEditorPages(): Type[];
19	
20	Description
21	Gets the set of component editor pages owned by the designer.
22	Return Value: The pages owned by the designer.
23	DataListDesigner class (System.Web.UI.Design.WebControls)
24	ToString
25	

1	
2	
3	Description
4	Provides design-time support for the
5	System.Web.UI.WebControls.DataList Web server control.
6	DataListDesigner
7	Example Syntax:
8	ToString
9	
10	[C#] public DataListDesigner();
11	[C++] public: DataListDesigner();
12	[VB] Public Sub New()
13	[JScript] public function DataListDesigner();
14	
15	Description
16	Initializes a new instance of the
17	System.Web.UI.Design.WebControls.DataListDesigner class
18	ActiveTemplateEditingFrame
19	AllowResize
20	ToString
21	
22	
23	Description
24	Indicates whether the data list can be resized.
25	AssociatedComponents

1	Behavior
2	CanEnterTemplateMode
3	Component
4	DataBindings
5	DataKeyField
6	DataMember
7	DataSource
8	DesignTimeElement
9	DesignTimeElementView
10	DesignTimeHtmlRequiresLoadComplete
11	HidePropertiesInTemplateMode
12	ID
13	InheritanceAttribute
14	Inherited
15	InTemplateMode
16	IsDirty
17	ReadOnly
18	ShadowProperties
19	ShouldCodeSerialize
20	TemplatesExist
21	ToString
22	
23	
24	Description

Indicates whether templates associated to the designer currently exist.

1	Verbs
2	CreateTemplateEditingFrame
3	
4	[C#] protected override ITemplateEditingFrame
5	CreateTemplateEditingFrame(TemplateEditingVerb verb);
6	[C++] protected: ITemplateEditingFrame*
7	CreateTemplateEditingFrame(TemplateEditingVerb* verb);
8	[VB] Overrides Protected Function CreateTemplateEditingFrame(ByVal verb As
9	TemplateEditingVerb) As ITemplateEditingFrame
10	[JScript] protected override function CreateTemplateEditingFrame(verb:
11	TemplateEditingVerb): ITemplateEditingFrame;
12	
13	Description
14	Creates a template editing frame using the specified verb.
15	Return Value: An System.Web.UI.Design.ITemplateEditingFrame. The verb
16	that was invoked to create a template editing frame.
17	Dispose
18	
19	[C#] protected override void Dispose(bool disposing);
20	[C++] protected: void Dispose(bool disposing);
21	[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)
22	[JScript] protected override function Dispose(disposing : Boolean);
23	
24	Description
25	

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Releases the unmanaged resources used by the System.Web.UI.Design.WebControls.DataListDesigner and optionally releases the managed resources.

This method is called by the public **Dispose()** method and the **System.Object.Finalize** method. **true** to release both managed and unmanaged resources; **false** to release only unmanaged resources.

GetCachedTemplateEditingVerbs

[C#] protected override TemplateEditingVerb[]

GetCachedTemplateEditingVerbs();

[C++] protected: TemplateEditingVerb\* GetCachedTemplateEditingVerbs() [];

[VB] Overrides Protected Function GetCachedTemplateEditingVerbs() As

TemplateEditingVerb()

[JScript] protected override function GetCachedTemplateEditingVerbs():

TemplateEditingVerb[];

Description

Gets the cached template editing verbs available to this designer.

Return Value: The cached template editing verbs available to this designer.

**GetDesignTimeHtml** 

[C#] public override string GetDesignTimeHtml();

[C++] public: String\* GetDesignTimeHtml();

[VB] Overrides Public Function GetDesignTimeHtml() As String

[JScript] public override function GetDesignTimeHtml(): String;

	5
	6
	7
	8
	9
	10
	11
ind had the	12
	13
	14
	15
	16
	17
	18
	19
	20
	21

Description

3

Gets the HTML that is used to represent the control at design time.

Return Value: The HTML that is used to represent the control at design time.

GetEmptyDesignTimeHtml

[C#] protected override string GetEmptyDesignTimeHtml();

[C++] protected: String\* GetEmptyDesignTimeHtml();

[VB] Overrides Protected Function GetEmptyDesignTimeHtml() As String

[JScript] protected override function GetEmptyDesignTimeHtml(): String;

Description

Gets the HTML used to represent an empty template-based control at design time.

Return Value: The HTML used to represent an empty template-based control at design time.

GetErrorDesignTimeHtml

[C#] protected override string GetErrorDesignTimeHtml(Exception e);

[C++] protected: String\* GetErrorDesignTimeHtml(Exception\* e);

[VB] Overrides Protected Function GetErrorDesignTimeHtml(ByVal e As

Excéption) As String

[JScript] protected override function GetErrorDesignTimeHtml(e: Exception):

String;

23

24

25

lee@hayes pilc 509-324-9256 1421 *MS1-863US.APP* 

17

18

19

20

21

22

23

25

-			, •	
110	SCY	47 M	†1/	111
170		1.17	LLI	,,,

2

3

5

Gets the HTML to display at design-time for the specified exception when an error has been encountered while rendering the control.

Return Value: The HTML displayed at design-time for the specified exception.

The exception to display the error message for.

**GetTemplateContainerDataItemProperty** 

[C#] public override string GetTemplateContainerDataItemProperty(string templateName);

[C++] public: String\* GetTemplateContainerDataItemProperty(String\* templateName);

[VB] Overrides Public Function GetTemplateContainerDataItemProperty(ByVal templateName As String) As String

[JScript] public override function

Get Template Container Data Item Property (template Name: String): String;

## Description

Gets the template's container's data item property.

Return Value: The data item property of the container of the template. The name of the template.

GetTemplateContent

[C#] public override string GetTemplateContent(ITemplateEditingFrame editingFrame, string templateName, out bool allowEditing);

1	[C++] public: String* GetTemplateContent(ITemplateEditingFrame*
2	editingFrame, String* templateName, bool* allowEditing);
3	[VB] Overrides Public Function GetTemplateContent(ByVal editingFrame As
4	ITemplateEditingFrame, ByVal templateName As String, ByRef allowEditing As
5	Boolean) As String
6	[JScript] public override function GetTemplateContent(editingFrame:
7	ITemplateEditingFrame, templateName : String, allowEditing : Boolean) : String;
8	
9	Description
10	Gets the template's content.
11	Return Value: The template's content. The template frame to get content from. The
12	name of the template. A boolean variable that will be set to true if the template's
13	content can be edited, or false if the content is read-only.
14	Initialize
15	
16	[C#] public override void Initialize(IComponent component);
17	[C++] public: void Initialize(IComponent* component);
18	[VB] Overrides Public Sub Initialize(ByVal component As IComponent)
19	[JScript] public override function Initialize(component : IComponent);
20	
21	Description
22	Initializes the designer with the System.Web.UI.WebControls.DataList
23	control that this instance of the designer is associated with. The associated
24	System.Web.UI.WebControls.DataList control.
25	OnTemplateEditingVerbsChanged

1	
2	[C#] protected override void OnTemplateEditingVerbsChanged();
3	[C++] protected: void OnTemplateEditingVerbsChanged();
4	[VB] Overrides Protected Sub OnTemplateEditingVerbsChanged()
5	[JScript] protected override function OnTemplateEditingVerbsChanged();
6	
7	Description
8	Provides an opportunity to do processing or other actions when a change
9	has been made to the template editing verbs.
10	SetTemplateContent
11	
12	[C#] public override void SetTemplateContent(ITemplateEditingFrame
13	editingFrame, string templateName, string templateContent);
14	[C++] public: void SetTemplateContent(ITemplateEditingFrame* editingFrame,
15	String* templateName, String* templateContent);
16	[VB] Overrides Public Sub SetTemplateContent(ByVal editingFrame As
17	ITemplateEditingFrame, ByVal templateName As String, ByVal templateContent
18	As String)
19	[JScript] public override function SetTemplateContent(editingFrame:
20	ITemplateEditingFrame, templateName : String, templateContent : String);
21	
22	Description
23	Sets the template's content. The template frame to set content for. The name
24	of the template. The content to set.
25	HyperLinkDesigner class (System.Web.UI.Design.WebControls)

1	UpdateDesignTimeHtml
2	
3	
4	Description
5	Provides design-time support for the
6	System.Web.UI.WebControls.HyperLink Web server control.
7	HyperLinkDesigner
8	Example Syntax:
9	UpdateDesignTimeHtml
10	
11	[C#] public HyperLinkDesigner();
12	[C++] public: HyperLinkDesigner();
13	[VB] Public Sub New()
14	[JScript] public function HyperLinkDesigner();
15	AllowResize
16	AssociatedComponents
17	Behavior
18	Component
19	DataBindings
20	DesignTimeElement
21	DesignTimeElementView
22	DesignTimeHtmlRequiresLoadComplete
23	ID
24	InheritanceAttribute
25	Inherited

1	IsDirty
2	ReadOnly
3	ShadowProperties
4	ShouldCodeSerialize
5	Verbs
6	GetDesignTimeHtml
7	
8	[C#] public override string GetDesignTimeHtml();
9	[C++] public: String* GetDesignTimeHtml();
10	[VB] Overrides Public Function GetDesignTimeHtml() As String
11	[JScript] public override function GetDesignTimeHtml(): String;
12	
13	Description
14	Gets the HTML that is used to represent the control at design time.
15	Return Value: The HTML that is used to represent the control at design time.
16	LabelDesigner class (System.Web.UI.Design.WebControls)
17	UpdateDesignTimeHtml
18	
19	
20	Description
21	Provides design-time support for the System.Web.UI.WebControls.Label
22	Web server control.
23	LabelDesigner
24	Example Syntax:
25	UpdateDesignTimeHtml

1	
2	[C#] public LabelDesigner();
3	[C++] public: LabelDesigner();
4	[VB] Public Sub New()
5	[JScript] public function LabelDesigner();
6	AllowResize
7	AssociatedComponents
8	Behavior
9	Component
10	DataBindings
11	DesignTimeElement
12	DesignTimeElementView
13	DesignTimeHtmlRequiresLoadComplete
14	ID
15	InheritanceAttribute
16	Inherited
17	IsDirty
18	ReadOnly
19	ShadowProperties
20	ShouldCodeSerialize
21	Verbs
22	LinkButtonDesigner class (System.Web.UI.Design.WebControls)
23	UpdateDesignTimeHtml
24	
25	

1	
2	Description
3	Description
4	Provides design-time support for the
5	System.Web.UI.WebControls.LinkButton Web server control.
6	LinkButtonDesigner
7	Example Syntax:
8	UpdateDesignTimeHtml
9	
10	[C#] public LinkButtonDesigner();
11	[C++] public: LinkButtonDesigner();
12	[VB] Public Sub New()
13	[JScript] public function LinkButtonDesigner();
14	AllowResize
15	AssociatedComponents
16	Behavior
17	Component
18	DataBindings
19	DesignTimeElement
20	DesignTimeElementView
21	DesignTimeHtmlRequiresLoadComplete
22	ID
23	InheritanceAttribute
24	Inherited
25	IsDirty

1	ReadOnly
2	ShadowProperties
3	ShouldCodeSerialize
4	Verbs
5	ListControlDataBindingHandler class
6	(System.Web.UI.Design.WebControls)
7	UpdateDesignTimeHtml
8	
9	
10	Description
11	Provides a data binding handler for a
12	System.Web.UI.WebControls.ListControl.
13	ListControlDataBindingHandler
14	Example Syntax:
15	UpdateDesignTimeHtml
16	
17	[C#] public ListControlDataBindingHandler();
18	[C++] public: ListControlDataBindingHandler();
19	[VB] Public Sub New()
20	[JScript] public function ListControlDataBindingHandler();
21	DataBindControl
22	
23	[C#] public override void DataBindControl(IDesignerHost designerHost, Control
24	control);
25	[C++] public: void DataBindControl(IDesignerHost* designerHost, Control*

```
control);
    [VB] Overrides Public Sub DataBindControl(ByVal designerHost As
    IDesignerHost, ByVal control As Control)
    [JScript] public override function DataBindControl(designerHost: IDesignerHost,
    control: Control);
6
    Description
           Adds a data binding to the specified control. The designer host for the
8
    document that contains the control. The control to add this data binding to.
           ListControlDesigner class (System.Web.UI.Design.WebControls)
10
           ToString
11
12
13
    Description
14
           Provides design-time support for
15
    System.Web.UI.WebControls.ListControl Web server controls.
16
           ListControlDesigner
17
           Example Syntax:
18
           ToString
19
20
    [C#] public ListControlDesigner();
21
    [C++] public: ListControlDesigner();
22
    [VB] Public Sub New()
23
    [JScript] public function ListControlDesigner();
25
```

1	
2	Description
3	Initializes a new instance of the
4	System.Web.UI.Design.WebControls.ListControlDesigner class.
5	AllowResize
6	AssociatedComponents
7	Behavior
8	Component
9	DataBindings
10	DataMember
11	ToString
12	
13	
14	Description
15	Gets or sets the data member for the control.
16	DataSource
17	ToString
18	
19	[C#] public string DataSource {get; set;}
20	[C++] public:property String* get_DataSource();public:property void
21	set_DataSource(String*);
22	[VB] Public Property DataSource As String
23	[JScript] public function get DataSource() : String;public function set
24	DataSource(String);
25	

	11	
	1	
	2	Description
	3	Gets or sets the data source property of the control.
	4	DataTextField
	5	ToString
	6	
	7	[C#] public string DataTextField {get; set;}
	8	[C++] public:property String* get_DataTextField();public:property void
	9	<pre>set_DataTextField(String*);</pre>
And the think	10	[VB] Public Property DataTextField As String
, MIN.	11	[JScript] public function get DataTextField(): String; public function set
	12	DataTextField(String);
	13	
	14	Description
	15	Gets or sets the data text field of the control.
	16	DataValueField
	17	ToString
	18	
	19	[C#] public string DataValueField {get; set;}
	20	[C++] public:property String* get_DataValueField();public:property void
	21	set_DataValueField(String*);
	22	[VB] Public Property DataValueField As String
	23	[JScript] public function get DataValueField(): String;public function set
	24	DataValueField(String);
	25	
	•	

1	
2	Description
3	Gets or sets the data value field of the control.
4	DesignTimeElement
5	DesignTimeElementView
6	DesignTimeHtmlRequiresLoadComplete
7	ID
8	InheritanceAttribute
9	Inherited
10	IsDirty
11	ReadOnly
12	ShadowProperties
13	ShouldCodeSerialize
14	Verbs
15	GetDesignTimeHtml
16	
17	[C#] public override string GetDesignTimeHtml();
18	[C++] public: String* GetDesignTimeHtml();
19	[VB] Overrides Public Function GetDesignTimeHtml() As String
20	[JScript] public override function GetDesignTimeHtml(): String;
21	
22	Description
23	Gets the HTML that is used to represent the control at design time.
24	Return Value: The HTML that is used to represent the control at design time.
25	GetResolvedSelectedDataSource

1 [C#] public IEnumerable GetResolvedSelectedDataSource(); [C++] public: \_\_sealed IEnumerable\* GetResolvedSelectedDataSource(); 3 [VB] NotOverridable Public Function GetResolvedSelectedDataSource() As 4 **IEnumerable** 5 [JScript] public function GetResolvedSelectedDataSource(): IEnumerable; 7 Description 8 Gets the data member that is currently selected within the data source 9 currently bound to the control. 10 Return Value: The currently selected data member, or null if the control was not 11 bound to a data source, or the data source, site of the designer's component, or the 12 container of the data source could not be accessed. 13 GetSelectedDataSource 14 15 [C#] public object GetSelectedDataSource(); 16 [C++] public: sealed Object\* GetSelectedDataSource(); 17 [VB] NotOverridable Public Function GetSelectedDataSource() As Object 18 [JScript] public function GetSelectedDataSource() : Object; 19

Description

20

21

22

23

24

25

Gets the selected data source component from the component's container.

Return Value: The selected data source, or **null** if a data source is not found or if a data source with the same name does not exist.

Initialize

1	
2	[C#] public override void Initialize(IComponent component);
3	[C++] public: void Initialize(IComponent* component);
4	[VB] Overrides Public Sub Initialize(ByVal component As IComponent)
5	[JScript] public override function Initialize(component : IComponent);
6	
7	Description
8	Initializes the component for design.
9	This method is called to initialize the designer with the component to
10	design. The control that is being designed.
11	OnComponentChanged
12	
13	[C#] public override void OnComponentChanged(object source,
14	ComponentChangedEventArgs ce);
15	[C++] public: void OnComponentChanged(Object* source,
16	ComponentChangedEventArgs* ce);
17	[VB] Overrides Public Sub OnComponentChanged(ByVal source As Object,
18	ByVal ce As ComponentChangedEventArgs)
19	[JScript] public override function OnComponentChanged(source : Object, ce :
20	ComponentChangedEventArgs);
21	
22	Description
23	Raises the ComponentChanged event.
24	Handles changes made to the component. This includes changes made in
25	the Properties window. The source of the event. A

1	System.ComponentModel.Design.ComponentChangedEventArgs that
2	provides data about the event.
3	OnDataSourceChanged
4	
5	[C#] public virtual void OnDataSourceChanged();
6	[C++] public: virtual void OnDataSourceChanged();
7	[VB] Overridable Public Sub OnDataSourceChanged()
8	[JScript] public function OnDataSourceChanged();
9	
10	Description
11	Raises the DataSource event.
12	Handles changes made to the data source.
13	PreFilterProperties
14	
15	[C#] protected override void PreFilterProperties(IDictionary properties);
16	[C++] protected: void PreFilterProperties(IDictionary* properties);
17	[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As
18	IDictionary)
19	[JScript] protected override function PreFilterProperties(properties : IDictionary);
20	
21	Description
22	Filters the properties exposed through a
23	System.ComponentModel.TypeDescriptor and replaces the property descriptor
24	for the DataSource property, which contains the value used at runtime, with a
25	

	1	property descriptor that contains a value indicating a design-time data source. The
	2	properties of the control.
	3	ListItemsCollectionEditor class (System.Web.UI.Design.WebControls)
	4	UpdateDesignTimeHtml
	5	
	6	
	7	Description
	8	Provides a user interface for editing the items collection of a list.
	9	ListItemsCollectionEditor
in the	10	Example Syntax:
and and that the the their that the	11	UpdateDesignTimeHtml
	12	
	13	[C#] public ListItemsCollectionEditor(Type type);
	14	[C++] public: ListItemsCollectionEditor(Type* type);
	15	[VB] Public Sub New(ByVal type As Type)
	16	[JScript] public function ListItemsCollectionEditor(type: Type);
ļ L	17	
	18	Description
	19	Initializes a new instance of the
	20	System.Web.UI.Design.WebControls.ListItemsCollectionEditor class. The
	21	type of the collection to edit.
	22	CollectionItemType
	23	CollectionType
	24	Context
	25	HelpTopic

	1	NewItemTypes
	2	CanSelectMultipleInstances
	3	
	4	[C#] protected override bool CanSelectMultipleInstances();
	5	[C++] protected: bool CanSelectMultipleInstances();
	6	[VB] Overrides Protected Function CanSelectMultipleInstances() As Boolean
	7	[JScript] protected override function CanSelectMultipleInstances(): Boolean;
	8	
	9	Description
200	10	Indicates whether multiple items in the list can be selected at the same time
	11	Return Value: true if multiple items can be selected at the same time; otherwise,
	12	false .
	13	PanelDesigner class (System.Web.UI.Design.WebControls)
Ë	14	ToString
	15	
	16	
ud;	17	Description
	18	Provides design-time support for the System.Web.UI.WebControls.Panel
	19	Web server control.
	20	PanelDesigner
	21	Example Syntax:
	22	ToString
	23	
	24	[C#] public PanelDesigner();
	25	[C++] public: PanelDesigner();

	1	[VB] Public Sub New()			
	2	[JScript] public function PanelDesigner();			
	3	AllowResize			
	4	AssociatedComponents			
	5	Behavior			
	6	Component			
	7	DataBindings			
	8	DesignTimeElement			
	9	DesignTimeElementView			
	10	DesignTimeHtmlRequiresLoadComplete			
S	11	ID			
1	12	InheritanceAttribute			
	13	Inherited			
	14	IsDirty			
11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	15	ReadOnly			
***	16	ShadowProperties			
	17	ShouldCodeSerialize			
	18	Verbs			
	19	MapPropertyToStyle			
	20				
	21	[C#] protected override void MapPropertyToStyle(string propName, object			
	22	varPropValue);			
	23	[C++] protected: void MapPropertyToStyle(String* propName, Object*			
	24	varPropValue);			
	25	[VB] Overrides Protected Sub MapPropertyToStyle(ByVal propName As String,			

1	ByVal varPropValue As Object)
2	[JScript] protected override function MapPropertyToStyle(propName : String,
3	varPropValue : Object);
4	
5	Description
6	Maps a specified property and value to a specified HTML style. The
7	property name. The property value.
8	OnBehaviorAttached
9	
10	[C#] protected override void OnBehaviorAttached();
11	[C++] protected: void OnBehaviorAttached();
12	[VB] Overrides Protected Sub OnBehaviorAttached()
13	[JScript] protected override function OnBehaviorAttached();
14	
15	Description
16	Provides notification when a behavior is attached to the designer.
17	RegexEditorDialog class (System.Web.UI.Design.WebControls)
18	UpdateDesignTimeHtml
19	
20	
21	Description
22	Provides a dialog for editing regular expressions used by the
23	System.Web.UI.WebControls.RegularExpressionValidator.
24	RegexEditorDialog
25	Example Syntax:
	3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24

1	UpdateDesignTimeHtml			
2				
3	[C#] public RegexEditorDialog(ISite site);			
4	[C++] public: RegexEditorDialog(ISite* site);			
5	[VB] Public Sub New(ByVal site As ISite)			
6	[JScript] public function RegexEditorDialog(site : ISite);			
7				
8	Description			
9	Initializes a new instance of the			
System.Web.UI.Design.WebControls.RegexEditorDialog class. The site				
11	dialog.			
12	AcceptButton			
13	AccessibilityObject			
14	AccessibleDefaultActionDescription			
15	AccessibleDescription			
16	AccessibleName			
17	AccessibleRole			
18	ActiveControl			
19	ActiveMdiChild			
20	AllowDrop			
21	AllowTransparency			
22	Anchor			
23	AutoScale			
24	AutoScaleBaseSize			
25	AutoScroll			

25

DataBindings

AutoScrollMargin AutoScrollMinSize 2 AutoScrollPosition 3 BackColor BackgroundImage 5 Binding Context**Bottom** 7 Bounds 8 CancelButton 9 CanFocus 10 CanSelect 11 Capture 12 CausesValidation 13 ClientRectangle 14 ClientSize 15 CompanyName 16 Container 17 ContainsFocus 18 ContextMenu 19 ControlBox 20 Controls 21 Created 22 CreateParams 23 Cursor

> 1442 MS1-863US.APP lee@hayes plic 509-324-9256

	2	DefaultSize		
	3	DesignMode		
	4	DesktopBounds		
	5	DesktopLocation		
	6	DialogResult		
	7	DisplayRectangl		
	8	Disposing		
	9	Dock		
i em	10	DockPadding		
	11	Enabled		
T. T.	12	Events		
	13	Focused		
C ·	14	Font		
	15	FontHeight		
201 201 201	16	ForeColor		
natif	17	FormBorderStyle		
	18	Handle		
	19	HasChildren		
	20	Height		
	21	HelpButton		
	22	HScroll		
	23	Icon		
	24	ImeMode		

InvokeRequired

DefaultImeMode

IsDisposed 2 IsHandleCreated 3 IsMdiChild IsMdiContainer 5 IsRestrictedWindow 6 KeyPreview 7 Left Location 9 MaximizeBox 10 MaximizedBounds 11 MaximumSize 12 MdiChildren 13 MdiParent 14 Menu 15 MergedMenu 16 MinimizeBox 17 MinimumSize 18 Modal 19 Name 20 Opacity 21 OwnedForms 22 Owner 23

Parent

ParentForm

24

25

IsAccessible

ProductName ProductVersion RecreatingHandle Region RegularExpression UpdateDesignTimeHtml Description Gets or sets the name of the regular expression to edit. 10 Render Right To Left11 ResizeRedraw 12 Right 13 Right To Left14 ShowFocusCues 15 ShowInTaskbar 16 ShowKeyboardCues 17 Site 18 Size 19 SizeGripStyle 20 StartPosition 21 TabIndex 22 TabStop 23 Tag 24 Text

Top TopLevel **TopLevelControl TopMost** TransparencyKey Visible 6 **VScroll** Width 8 WindowState 9 WindowTarget 10 cmdHelp Click 11 12 [C#] protected void cmdHelp Click(object sender, EventArgs e); 13 [C++] protected: void cmdHelp\_Click(Object\* sender, EventArgs\* e); 14 [VB] Protected Sub cmdHelp Click(ByVal sender As Object, ByVal e As 15 EventArgs) 16 [JScript] protected function cmdHelp Click(sender: Object, e: EventArgs); 18 Description 19 Represents the method that will handle the Help event of the dialog. The 20 source of the event. An EventArgs that provides data for the event. 21 cmdOK Click 22 23 [C#] protected void cmdOK\_Click(object sender, EventArgs e); 24 [C++] protected: void cmdOK\_Click(Object\* sender, EventArgs\* e);

1	[VB] Protected Sub cmdOK_Click(ByVal sender As Object, ByVal e As
2	EventArgs)
3	[JScript] protected function cmdOK_Click(sender : Object, e : EventArgs);
4	
5	Description
6	Represents the method that will handle the OK event of the dialog. The
7	source of the event. An EventArgs that provides data for the event.
8	cmdTestValidate_Click
9	
10	[C#] protected void cmdTestValidate_Click(object sender, EventArgs args);
11	[C++] protected: void cmdTestValidate_Click(Object* sender, EventArgs* args);
12	[VB] Protected Sub cmdTestValidate_Click(ByVal sender As Object, ByVal args
13	As EventArgs)
14	[JScript] protected function cmdTestValidate_Click(sender : Object, args :
15	EventArgs);
16	
17	Description
18	Represents the method that will handle the XXX event of a XXX. The
19	source of the event. An EventArgs that provides data for the event.
20	Dispose
21	
22	[C#] protected override void Dispose(bool disposing);
23	[C++] protected: void Dispose(bool disposing);
24	[VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean)
25	[JScript] protected override function Dispose(disposing : Boolean);

3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

T .	•	. •
Desc	arin	tran
17630		uu

Releases the unmanaged resources used by the

**System.Web.UI.Design.WebControls.RegexEditorDialog** and optionally releases the managed resources.

This method is called by the public **Dispose()** method and the **System.Object.Finalize** method.

1stStandardExpressions SelectedIndexChanged

[C#] protected void lstStandardExpressions\_SelectedIndexChanged(object sender, EventArgs e);

[C++] protected: void lstStandardExpressions\_SelectedIndexChanged(Object\* sender, EventArgs\* e);

[VB] Protected Sub lstStandardExpressions\_SelectedIndexChanged(ByVal sender As Object, ByVal e As EventArgs)

[JScript] protected function

lstStandardExpressions\_SelectedIndexChanged(sender : Object, e : EventArgs);

## Description

Represents the method that will handle the XXX event of a XXX. The source of the event. An EventArgs that provides data for the event.

RegexTypeEditor\_Activated

[C#] protected void RegexTypeEditor\_Activated(object sender, EventArgs e);

[C++] protected: void RegexTypeEditor\_Activated(Object\* sender, EventArgs\*

1	e);
2	[VB] Protected Sub RegexTypeEditor_Activated(ByVal sender As Object, ByVal
3	e As EventArgs)
4	[JScript] protected function RegexTypeEditor_Activated(sender : Object, e :
5	EventArgs);
6	
7	Description
8	Represents the method that will handle the XXX event of a XXX. The
9	source of the event. An EventArgs that provides data for the event.
10	txtExpression_TextChanged
11	·.
12	[C#] protected void txtExpression_TextChanged(object sender, EventArgs e);
13	[C++] protected: void txtExpression_TextChanged(Object* sender, EventArgs*
14	e);
15	[VB] Protected Sub txtExpression_TextChanged(ByVal sender As Object, ByVal
16	e As EventArgs)
17	[JScript] protected function txtExpression_TextChanged(sender : Object, e :
18	EventArgs);
19	
20	Description
21	Represents the method that will handle the XXX event of a XXX. The
22	source of the event. An EventArgs that provides data for the event.
23	RegexTypeEditor class (System.Web.UI.Design.WebControls)
24	WndProc
25	

	1	
	2	
	3	Description
	4	Provides a user interface for editing regular expressions.
	5	RegexTypeEditor
	6	Example Syntax:
	7	WndProc
	8	
	9	[C#] public RegexTypeEditor();
1000 a	10	[C++] public: RegexTypeEditor();
	11	[VB] Public Sub New()
	12	[JScript] public function RegexTypeEditor();
	13	EditValue
	14	
	15	[C#] public override object EditValue(ITypeDescriptorContext context,
	16	IServiceProvider provider, object value);
जारी श्रीद	17	[C++] public: Object* EditValue(ITypeDescriptorContext* context,
	18	IServiceProvider* provider, Object* value);
	19	[VB] Overrides Public Function EditValue(ByVal context As
	20	ITypeDescriptorContext, ByVal provider As IServiceProvider, ByVal value As
	21	Object) As Object
	22	[JScript] public override function EditValue(context : ITypeDescriptorContext,
	23	provider : IServiceProvider, value : Object) : Object;
	24	
	25	Description
		II and the state of the state o

2

3

5

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

25

Edits the value of the specified object using the specified service provider and context.

Return Value: The new value of the object. If the value of the object hasn't

changed, this should return the same object it was passed. An

System.ComponentModel.ITypeDescriptorContext that can provide additional

System.ComponentModel.ITypeDescriptorContext that can provide additional context information. A service provider. The object to edit the value of.

GetEditStyle

[C#] public override UITypeEditorEditStyle

 $GetEditStyle (ITypeDescriptorContext\ context);\\$ 

[C++] public: UITypeEditorEditStyle GetEditStyle(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetEditStyle(ByVal context As

ITypeDescriptorContext) As UITypeEditorEditStyle

[JScript] public override function GetEditStyle(context : ITypeDescriptorContext): UITypeEditorEditStyle;

Description

Gets the editor style used by the

System.Web.UI.Design.WebControls.RegexTypeEditor.EditValue(System.ComponentModel.ITypeDescriptorContext,System.IServiceProvider,System.Object) method.

Return Value: A System.Drawing.Design.UITypeEditorEditStyle that indicates the editor style. An System.ComponentModel.ITypeDescriptorContext that may be used to gain additional context information.

1	RepeaterDesigner class (System.Web.UI.Design.WebControls)
2	ToString
3	
4	
5	Description
6	Provides a designer for the System.Web.UI.WebControls.Repeater
7	control.
8	RepeaterDesigner
9	Example Syntax:
10	ToString
11	
12	[C#] public RepeaterDesigner();
13	[C++] public: RepeaterDesigner();
14	[VB] Public Sub New()
15	[JScript] public function RepeaterDesigner();
16	
17	Description
18	Initializes a new instance of the
19	System.Web.UI.Design.WebControls.RepeaterDesigner class.
20	AllowResize
21	AssociatedComponents
22	Behavior
23	Component
24	DataBindings
25	DataMember

	1	ToString
	2	
	3	
	4	Description
	5	
	6	DataSource
	7	ToString
	8	
	9	[C#] public string DataSource {get; set;}
	10	[C++] public:property String* get_DataSource();public:property void
find that the first time to be the first time to be	11	set_DataSource(String*);
	12	[VB] Public Property DataSource As String
And And	13	[JScript] public function get DataSource(): String;public function set
i indi	14	DataSource(String);
er II'' dime Truff	15	
, mu, , mu,	16	Description
•	17	Designer implementation of DataSource property that operates on the
	18	DataSource property in the control's binding collection.
	19	DesignTimeElement
	20	DesignTimeElementView
	21	DesignTimeHtmlRequiresLoadComplete
	22	ID
	23	InheritanceAttribute
	24	Inherited
	25	IsDirty

ReadOnly **ShadowProperties** 2 ShouldCodeSerialize 3 **Templates**Exist **ToString** 7 Description 8 Verbs 9 Dispose 10 11 [C#] protected override void Dispose(bool disposing); 12 [C++] protected: void Dispose(bool disposing); 13 [VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean) 14 [JScript] protected override function Dispose(disposing : Boolean); 15 16 Description 17 Performs the cleanup of the designer class. 18 GetDesignTimeDataSource 19 20 [C#] protected IEnumerable GetDesignTimeDataSource(int minimumRows); 21 [C++] protected: IEnumerable\* GetDesignTimeDataSource(int minimumRows); 22 [VB] Protected Function GetDesignTimeDataSource(ByVal minimumRows As 23 Integer) As IEnumerable 24 [JScript] protected function GetDesignTimeDataSource(minimumRows: int): 25

IEnumerable;

Description

Returns a sample data matching the schema of the selected datasource.

Return Value: A live datasource for use at design-time. The minimum rows of sample data the datasource should contain.

GetDesignTimeDataSource

[C#] protected IEnumerable GetDesignTimeDataSource(IEnumerable selectedDataSource, int minimumRows);

[C++] protected: IEnumerable\* GetDesignTimeDataSource(IEnumerable\* selectedDataSource, int minimumRows);

[VB] Protected Function GetDesignTimeDataSource(ByVal selectedDataSource
As IEnumerable, ByVal minimumRows As Integer) As IEnumerable
[JScript] protected function GetDesignTimeDataSource(selectedDataSource:
IEnumerable, minimumRows: int): IEnumerable;

Description

Returns a sample data matching the schema of the selected datasource.

Return Value: A live datasource for use at design-time. The selected datasource to be used as a reference for the shape of the data. The minimumn rows of sample data the datasource should contain.

GetDesignTimeHtml

[C#] public override string GetDesignTimeHtml();

```
[C++] public: String* GetDesignTimeHtml();
    [VB] Overrides Public Function GetDesignTimeHtml() As String
    [JScript] public override function GetDesignTimeHtml(): String;
    Description
           Retrieves the HTML to be used for the design-time representation of the
6
    control.
    Return Value: Design Time HTML.
           GetEmptyDesignTimeHtml
9
10
    [C#] protected override string GetEmptyDesignTimeHtml();
11
    [C++] protected: String* GetEmptyDesignTimeHtml();
    [VB] Overrides Protected Function GetEmptyDesignTimeHtml() As String
13
    [JScript] protected override function GetEmptyDesignTimeHtml(): String;
14
15
    Description
16
           GetErrorDesignTimeHtml
17
18
    [C#] protected override string GetErrorDesignTimeHtml(Exception e);
19
    [C++] protected: String* GetErrorDesignTimeHtml(Exception* e);
20
    [VB] Overrides Protected Function GetErrorDesignTimeHtml(ByVal e As
21
    Exception) As String
22
    [JScript] protected override function GetErrorDesignTimeHtml(e: Exception):
23
    String;
24
25
```

Initialize

2	Description
3	GetResolvedSelectedDataSource
4	
5	[C#] public IEnumerable GetResolvedSelectedDataSource();
6	[C++] public:sealed IEnumerable* GetResolvedSelectedDataSource();
7	[VB] NotOverridable Public Function GetResolvedSelectedDataSource() As
8	IEnumerable
9	[JScript] public function GetResolvedSelectedDataSource(): IEnumerable;
10	
11	Description
12	GetSelectedDataSource
13	
14	[C#] public object GetSelectedDataSource();
15	[C++] public:sealed Object* GetSelectedDataSource();
16	[VB] NotOverridable Public Function GetSelectedDataSource() As Object
17	[JScript] public function GetSelectedDataSource(): Object;
18	
19	Description
20	Retrieves the selected datasource component from the component's
21	container.
22	Return Value: The selected datasource; null if a datasource is not found, or a
23	datasource with the same name does not exist.

1	
2	[C#] public override void Initialize(IComponent component);
3	[C++] public: void Initialize(IComponent* component);
4	[VB] Overrides Public Sub Initialize(ByVal component As IComponent)
5	[JScript] public override function Initialize(component : IComponent);
6	
7	Description
8	Initializes the designer with the Repeater control that this instance of the
9	designer is associated with. The associated Repeater control.
10	OnComponentChanged
11	
12	[C#] public override void OnComponentChanged(object source,
13	ComponentChangedEventArgs ce);
14	[C++] public: void OnComponentChanged(Object* source,
15	ComponentChangedEventArgs* ce);
16	[VB] Overrides Public Sub OnComponentChanged(ByVal source As Object,
17	ByVal ce As ComponentChangedEventArgs)
18	[JScript] public override function OnComponentChanged(source : Object, ce :
19	ComponentChangedEventArgs);
20	
21	Description
22	Handles changes made to the component. This includes changes made in
23	the Properties window.
24	OnDataSourceChanged
25	

	1	
	2	[C#] public virtual void OnDataSourceChanged();
	3	[C++] public: virtual void OnDataSourceChanged();
	4	[VB] Overridable Public Sub OnDataSourceChanged()
	5	[JScript] public function OnDataSourceChanged();
	6	
	7	Description
	8	Handles changes made to the data source Handles changes made to the data
	9	source
	10	PreFilterProperties
	11	
	12	[C#] protected override void PreFilterProperties(IDictionary properties);
	13	[C++] protected: void PreFilterProperties(IDictionary* properties);
	14	[VB] Overrides Protected Sub PreFilterProperties(ByVal properties As
	15	IDictionary)
TOTAL STATE OF THE	16	[JScript] protected override function PreFilterProperties(properties : IDictionary);
To Table	17	
	18	Description
	19	Filter the properties to replace the runtime DataSource property descriptor
	20	with the designer's.
	21	TableCellsCollectionEditor class (System.Web.UI.Design.WebControls)
	22	UpdateDesignTimeHtml
	23	
	24	
	25	Description

Provides a user interface for editing cells in a table. **TableCellsCollectionEditor** 2 Example Syntax: **UpdateDesignTimeHtml** 5 [C#] public TableCellsCollectionEditor(Type type); [C++] public: TableCellsCollectionEditor(Type\* type); [VB] Public Sub New(ByVal type As Type) 8 [JScript] public function TableCellsCollectionEditor(type: Type); 10 Description 11 Initializes a new instance of the 12 System.Web.UI.Design.WebControls.TableCellsCollectionEditor class. The 13 type of the collection to edit. 14 CollectionItemType 15 CollectionType 16 Context 17 HelpTopic 18 NewItemTypes 19 CanSelectMultipleInstances 20 21 [C#] protected override bool CanSelectMultipleInstances(); 22 [C++] protected: bool CanSelectMultipleInstances(); 23 [VB] Overrides Protected Function CanSelectMultipleInstances() As Boolean 24 [JScript] protected override function CanSelectMultipleInstances(): Boolean; 25

Description Indicates whether multiple table cells can be selected at the same time. 3 Return Value: true if multiple cells can be selected at the same time; otherwise, false. CreateInstance 6 7 [C#] protected override object CreateInstance(Type itemType); 8 [C++] protected: Object\* CreateInstance(Type\* itemType); [VB] Overrides Protected Function CreateInstance(ByVal itemType As Type) As 10 Object [JScript] protected override function CreateInstance(itemType: Type): Object; 12 13 Description 14 Creates an instance of the editor for use with the specified type. 15 Return Value: An object of the specified type. The type to create an instance of. 16 TableDesigner class (System.Web.UI.Design.WebControls) 17 **ToString** 18 19 20 Description 21 Provides design-time support for the System.Web.UI.WebControls.Table 22 Web server control. 23 **TableDesigner** 24 Example Syntax: 25

1	ToString
2	
3	[C#] public TableDesigner();
4	[C++] public: TableDesigner();
5	[VB] Public Sub New()
6	[JScript] public function TableDesigner();
7	AllowResize
8	AssociatedComponents
9	Behavior
10	Component
11	DataBindings
12	DesignTimeElement
13	DesignTimeElementView
14	DesignTimeHtmlRequiresLoadComplete
15	ID
16	InheritanceAttribute
17	Inherited
18	IsDirty
19	ReadOnly
20	ShadowProperties
21	ShouldCodeSerialize
22	Verbs
23	GetDesignTimeHtml
24	
25	[C#] public override string GetDesignTimeHtml();
	2 3 4 4 5 6 7 8 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

	1	[C++] public: String* GetDesignTimeHtml();
	2	[VB] Overrides Public Function GetDesignTimeHtml() As String
	3	[JScript] public override function GetDesignTimeHtml(): String;
	4	
	5	Description
	6	Gets the HTML that is used to represent the control at design time.
	7	Return Value: The HTML that is used to represent the control at design time.
	8	TableRowsCollectionEditor class (System.Web.UI.Design.WebControls)
	9	UpdateDesignTimeHtml
** <u>*</u>	10	
	11	
Tank hadi tanif tana tanif tanif tanif tanif	12	Description
A Sunt Sun	13	Provides a user interface for editing rows of a table.
Mr. thum fant's fan	14	TableRowsCollectionEditor
	15	Example Syntax:
ind ind	16	UpdateDesignTimeHtml
	17	
	18	[C#] public TableRowsCollectionEditor(Type type);
	19	[C++] public: TableRowsCollectionEditor(Type* type);
	20	[VB] Public Sub New(ByVal type As Type)
	21	[JScript] public function TableRowsCollectionEditor(type : Type);
	22	
	23	Description
	24	
	25	

1	Initializes a new instance of the
2	$System. Web. UI. Design. Web Controls. Table Rows Collection Editor\ class.\ The$
3	type of the collection to edit.
4	CollectionItemType
5	CollectionType
6	Context
7	HelpTopic
8	NewItemTypes
9	CanSelectMultipleInstances
10	
11	[C#] protected override bool CanSelectMultipleInstances();
12	[C++] protected: bool CanSelectMultipleInstances();
13	[VB] Overrides Protected Function CanSelectMultipleInstances() As Boolean
14	[JScript] protected override function CanSelectMultipleInstances(): Boolean;
15	
16	Description
17	Indicates whether multiple instances may be selected.
18	Return Value: true if multiple items can be selected at once; otherwise, false.
19	This implementation always returns false.
20	CreateInstance
21	
22	[C#] protected override object CreateInstance(Type itemType);
23	[C++] protected: Object* CreateInstance(Type* itemType);
24	[VB] Overrides Protected Function CreateInstance(ByVal itemType As Type) As
25	Object

	1	[JScript] protected override function CreateInstance(itemType : Type) : Object;						
	2							
	3	Description						
	4	Creates an instance of the specified type.						
	5	Return Value: An object of the specified type. The type to create an instance of.						
	6	XmlDesigner class (System.Web.UI.Design.WebControls)						
	7	ToString						
	8							
	9							
<b>2</b>	10	Description						
	11	Provides a designer for the System.Web.UI.WebControls.Xml Web						
	12	server control.						
	13	XmlDesigner						
	14	Example Syntax:						
that the first that the	15	ToString						
	16							
ii L	17	[C#] public XmlDesigner();						
	18	[C++] public: XmlDesigner();						
	19	[VB] Public Sub New()						
	20	[JScript] public function XmlDesigner();						
	21							
	22	Description						
	23	Initializes a new instance of the						
	24	System.Web.UI.Design.WebControls.XmlDesigner class.						
	25	AllowResize						

AssociatedComponents Behavior 2 Component 3 DataBindings DesignTimeElement 5 DesignTimeElementView 6 DesignTimeHtmlRequiresLoadComplete 7 ID 8 InheritanceAttribute 9 Inherited 10 **IsDirty** 11 ReadOnly 12 **ShadowProperties** 13 ShouldCodeSerialize 14 Verbs 15 Dispose 16 17 [C#] protected override void Dispose(bool disposing); 18 [C++] protected: void Dispose(bool disposing); 19 [VB] Overrides Protected Sub Dispose(ByVal disposing As Boolean) 20 [JScript] protected override function Dispose(disposing : Boolean); 21 22 Description 23 Performs the cleanup of the designer class. 24 GetDesignTimeHtml 25

1	
2	[C#] public override string GetDesignTimeHtml();
3	[C++] public: String* GetDesignTimeHtml();
4	[VB] Overrides Public Function GetDesignTimeHtml() As String
5	[JScript] public override function GetDesignTimeHtml(): String;
6	
7	Description
8	Gets the HTML that is used to represent the control at design time.
9	Return Value: The HTML that is used to represent the control at design time.
10	GetEmptyDesignTimeHtml
11	
12	[C#] protected override string GetEmptyDesignTimeHtml();
13	[C++] protected: String* GetEmptyDesignTimeHtml();
14	[VB] Overrides Protected Function GetEmptyDesignTimeHtml() As String
15	[JScript] protected override function GetEmptyDesignTimeHtml(): String;
16	
17	Description
18	Gets the HTML that is used to fill an empty control.
19	Return Value: The HTML used to fill an empty control.
20	Initialize
21	
22	[C#] public
23	
24	
25	

#### System.Web.UI.HtmlControls

# Description

The System.Web.UI.HtmlControls namespace is a collection of classes that allow you to create HTML server controls on a Web page. HTML server controls run on the server and map directly to standard HTML tags supported by all browsers. This allows you to programmatically control the HTML elements on the Web page.

HtmlAnchor class (System.Web.UI.HtmlControls)

### Description

Defines the methods, properties, and events for the System.Web.UI.HtmlControls.HtmlAnchor control. This class allows programmatic access to the HTML tag on the server.

There are two ways to use the System.Web.UI.HtmlControls.HtmlAnchor class. The first is for navigation: using the System.Web.UI.HtmlControls.HtmlAnchor.HRef property to define the location of the page to link to. The second is for postback events: using the System.Web.UI.HtmlControls.HtmlAnchor.ServerClick event to programmatically handle the user's click on a link.

Constructors:

HtmlAnchor

Example Syntax: 2 HtmlAnchor(); public [C#] HtmlAnchor(); public: [C++]New() Sub **Public** [VB] HtmlAnchor(); function public [JScript] 7 Description 8 the of instance **Initializes** new a 9  $System. Web. UI. Html Controls. Html Anchor\ class.$ 10 Properties: 11 Attributes 12 ChildControlsCreated 13 ClientID 14 Context 15 Controls 16 Disabled 17 EnableViewState 18 **Events** 19 HasChildViewState 20 HRef 21 22 23 Description 24 25

Gets or sets the URL target of the link specified in the System.Web.UI.HtmlControls.HtmlAnchor server control.

Use this property to specify the URL to link to when the System. Web.UI. Html Controls. Html Anchor is clicked.

ID

1

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

InnerHtml

InnerText

IsTrackingViewState

Name

#### Description

Gets or sets the bookmark name defined in the System. Web. UI. Html Controls. Html Anchor server control.

Use this property to mark sections on a Web page with a name. This allows you to link to this section from anywhere on the same page. For example, you can provide a table of contents at the top of a page that will link directly to topics on the page.

NamingContainer

Page

Parent

Site

Style

TagName

**Target** 

# Description Gets

Gets or sets the target window or frame to load Web page content into.

Use this property to specify the frame or window that displays the Web page linked to.

**TemplateSourceDirectory** 

Title

# Description

Gets or sets the title that the browser displays for a Web page.

Use this property to specify a custom title when the browser identifies the page you link to. This property also may be used to provide a custom tool tip for the System.Web.UI.HtmlControls.HtmlAnchor control.

UniqueID

ViewState

ViewStateIgnoresCase

Visible

# Description

Occurs on the server when a user clicks the System. Web. UI. Html Controls. Html Anchor control on the browser.

l,	355 155 155
١,	
ı,	
1	
ľ	ij
<u>;</u> ;	
1,	
11	
i,	
; ;	
ļ.,	sii.
ļ,	
1	## ##
	, i

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

when the raised is This event System. Web. UI. Html Controls. Html Anchor is clicked. Methods: **OnServerClick** OnServerClick(EventArgs e); virtual void protected [C#] OnServerClick(EventArgs\* e); void virtual [C++]protected: [VB] Overridable Protected Sub OnServerClick(ByVal e As EventArgs) OnServerClick(e EventArgs); function protected [JScript] Description System.Web.UI.HtmlControls.HtmlAnchor.ServerClick Raises the event. This allows you to handle the event directly. This server event causes a roundtrip to occur from the client to the server and back. An System. EventArgs that contains event data. RenderAttributes [C#] protected override void RenderAttributes(HtmlTextWriter writer); RenderAttributes(HtmlTextWriter\* writer); void protected: [C++]RenderAttributes(ByVal writer **Overrides** Protected Sub HtmlTextWriter) RenderAttributes(writer override function [JScript] protected HtmlTextWriter);

Description

#### IPostBackEventHandler.RaisePostBackEvent

2	

3

4

5

6

7

9

10

11 12

14

13

16

17

15

18 19

20 21

22

23

24

[C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);

[C++] void IPostBackEventHandler::RaisePostBackEvent(String\* eventArgument);

[VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements IPostBackEventHandler.RaisePostBackEvent

[JScript] function

IPostBackEventHandler.RaisePostBackEvent(eventArgument: String);

HtmlButton class (System.Web.UI.HtmlControls)

**TrackViewState** 

## Description

Defines the methods, properties, and events for the System.Web.UI.HtmlControls.HtmlButton control. This class allows programmatic access to the HTML tag on the server.

The element allows Web developers to create UI form buttons that can be composed of embedded HTML elements, including other server controls.

HtmlButton

Example Syntax:

**TrackViewState** 

[C#] public

HtmlButton();

HtmlButton(); public: [C++]New() Sub **Public** [VB] 2 HtmlButton(); function public [JScript] 4 Description the instance of **Initializes** new a 6 System. Web. UI. Html Controls. Html Button class. 7 Attributes 8 CausesValidation 9 TrackViewState 10 11 12 Description 13 Gets or sets a value indicating whether validation is performed when 14  $the\ System. Web. UI. Html Controls. Html Button\ control\ is\ clicked.$ 15 performed when an validation is default, page By 16 System.Web.UI.HtmlControls.HtmlButton control is clicked. Page validation 17 determines whether the input controls associated with a validation control on 18 the page all pass the validation rules specified by the validation control. 19 ChildControlsCreated 20 ClientID 21 Context 22 Controls 23 Disabled 24 EnableViewState 25

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

**Events** HasChildViewState ID InnerHtml InnerText IsTrackingViewState NamingContainer Page Parent Site Style TagName TemplateSourceDirectory UniqueID ViewState ViewStateIgnoresCase Visible TrackViewState Description the when user Occurs

Occurs when the user clicks an System.Web.UI.HtmlControls.HtmlButton control on the client Web page.

This event is raised when the System.Web.UI.HtmlControls.HtmlButton is clicked.

protected

protected:

protected

2

3

[C#]

[C++]

[JScript]

Description

6

8

9 10 11

12 13

15 16

14

17

18

19 20

21

24

25

22 23

7

System.Web.UI.HtmlControls.HtmlButton.ServerClick Raises the event. This allows you to handle the event directly.

virtual

virtual

function

This event causes a roundtrip to occur from the client to the server and back. It is deliberately different from the client-side OnClick event. In the with code a conflict exists between run event that a System.Web.UI.HtmlControls.HtmlButton.ServerClick event and code run by a client-side OnClick event, the server-side event instructions will override the client-side code. An System. EventArgs that contains the event data.

void

void

[VB] Overridable Protected Sub OnServerClick(ByVal e As EventArgs)

OnServerClick(e

OnServerClick(EventArgs

OnServerClick(EventArgs\*

e);

e);

EventArgs);

RenderAttributes

[C#] protected override void RenderAttributes(HtmlTextWriter writer); protected: RenderAttributes(HtmlTextWriter\* writer); void [C++]RenderAttributes(ByVal writer Sub **Overrides** Protected [VB] HtmlTextWriter)

RenderAttributes(writer [JScript] protected override function HtmlTextWriter);

1476 MS1-863US.APP lee@hayes plic 509+324+9256

1	
2	Description
3	IPostBackEventHandler.RaisePostBackEvent
4	
5	[C#] void IPostBackEventHandler.RaisePostBackEvent(string
6	eventArgument);
7	[C++] void IPostBackEventHandler::RaisePostBackEvent(String*
8	eventArgument);
9	[VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements
10	IPostBackEventHandler.RaisePostBackEvent
11	[JScript] function
12	IPostBackEventHandler.RaisePostBackEvent(eventArgument : String);
13	HtmlContainerControl class (System.Web.UI.HtmlControls)
14	TrackViewState
15	
16	
17	Description
18	Defines the methods, properties, and events available to all HTML
19	server controls that must have a closing tag.
20	The most common controls with a closing tag are the
21	Top of Form
22	
23	,,
24	,, and elements.
25	HtmlContainerControl

1	Example Syntax:								
2	TrackViewState								
3									
4	[C#]		public		H	ItmlCo	ntainerCo	ontrol();	
5	[C++]		public:		HtmlContainerControl();				
6	[VB]		Public		Sub			New()	
7	[JScript] pu	ıblic functio	n HtmlCont	ainerContro	ol(); Initialize	es a nev	w instanc	e of the	
8	System.Wo	eb.UI.Html	Controls.Ht	mlContain	erControl			class.	
9									
10	Description	ı							
11	Initi	alizes	a	new	instance		of	the	
12	System.W	eb.UI.Html	Controls.Ht	tmlContair	erControl	class	using	default	
13	values.								
14	Use	this const	ructor to	create and	initialize a	new	instance	of the	
15	System.W	eb.UI.Html	Controls.H	tmlContair	erControl	class	using	default	
16	values.								
17	Htm	nlContainer(	Control						
18	Exa	mple Syntax	:						
19	TrackViewState								
20									
21	[C#] public			HtmlContainerControl(string tag);					
22	[C++]	publi	c:	HtmlCon	ainerControl	l(String		tag);	
23	[VB]	Public	Sub	New(By			As	String)	
24	[JScript]	public	function	HtmlCo	ontainerConti	rol(tag	:	String);	
25									

17

18

19

20

21

22

23

24

25

Description

1

2

Initializes a new instance of the System. Web. UI. Html Controls. Html Container Control class using the specified tag name.

Use this constructor to create and initialize a new instance of the System.Web.UI.HtmlControls.HtmlContainerControl class using the specified tag. A string that specifies the tag name of the control.

Attributes

ChildControlsCreated

ClientID

Context

Controls

Disabled

EnableViewState

**Events** 

HasChildViewState

ID

InnerHtml

TrackViewState

Description

Gets or sets the content found between the opening and closing tags of the specified HTML server control.

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

the Use System.Web.UI.HtmlControls.HtmlContainerControl.InnerHtml property to programmatically modify the contents within the opening and closing tags of an HTML server control. InnerText **TrackViewState** InnerText {get; set;} string virtual public [C#] [C++] public: \_\_property virtual String\* get\_InnerText();public: \_\_property set InnerText(String\*); void virtual InnerText As String Public **Property** Overridable [VB] [JScript] public function get InnerText() : String;public function set InnerText(String); Description Gets or sets the text between the opening and closing tags of the specified HTML server control. the Use System.Web.UI.HtmlControls.HtmlContainerControl.InnerText property to programmatically modify the contents between the opening and closing tags of an HTML server control. IsTrackingViewState NamingContainer Page Parent

Site Style 2 TagName 3 **TemplateSourceDirectory** UniqueID 5 ViewState 6 ViewStateIgnoresCase 7 Visible 8 CreateControlCollection 9 10 CreateControlCollection(); ControlCollection override protected [C#] 11 CreateControlCollection(); ControlCollection\* protected: [C++]12 CreateControlCollection() As Overrides Protected Function [VB] 13 ControlCollection CreateControlCollection() function override [JScript] protected 15 ControlCollection; 16 LoadViewState 17 18 savedState); override void LoadViewState(object [C#] protected 19 savedState); LoadViewState(Object\* void protected: [C++]20 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object) 21 [JScript] protected override function LoadViewState(savedState : Object); 22 23 Description 24 Render 25

2	[C#] protected override void Render(HtmlTextWriter writer);		
3	[C++] protected: void Render(HtmlTextWriter* writer);		
4	[VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)		
5	[JScript] protected override function Render(writer : HtmlTextWriter);		
6			
7	Description		
8	RenderAttributes		
9			
10	[C#] protected override void RenderAttributes(HtmlTextWriter writer);		
11	[C++] protected: void RenderAttributes(HtmlTextWriter* writer);		
12	[VB] Overrides Protected Sub RenderAttributes(ByVal writer As		
13	HtmlTextWriter)		
14	[JScript] protected override function RenderAttributes(writer : HtmlTextWriter);		
15			
16	Description		
17	RenderEndTag		
18			
19	[C#] protected virtual void RenderEndTag(HtmlTextWriter writer);		
20	[C++] protected: virtual void RenderEndTag(HtmlTextWriter* writer);		
21	[VB] Overridable Protected Sub RenderEndTag(ByVal writer As		
22	HtmlTextWriter)		
23	[JScript] protected function RenderEndTag(writer : HtmlTextWriter);		
24			
25	Description		

MS1-863US.APP lee@hayes plic 509-324-9256

HtmlControl class (System.Web.UI.HtmlControls) **TrackViewState** 2 3 4 Description 5 Defines the methods, properties, and events common to all HTML server 6 controls in the Web Forms page framework. 7 The System.Web.UI.HtmlControls.HtmlControl class provides common 8 properties inherited by all HTML server control classes. It is not instantiated 9 directly. 10 HtmlControl 11 Example Syntax: 12 TrackViewState 13 14 [C#] public HtmlControl(); 15 [C++]public: HtmlControl(); 16 Sub [VB] Public New() 17 [JScript] public function HtmlControl(); Initializes a new instance of the 18 System.Web.UI.HtmlControls.HtmlControl class. 19 20 Description 21 **Initializes** of instance the a new 22 **System.Web.UI.HtmlControls.HtmlControl** class using default values. 23 This constructor is used to create and initialize a new instance of the 24 System. Web. UI. Html Controls. Html Control class using default values. 25

HtmlControl 1 Example Syntax: 2 TrackViewState 3 HtmlControl(string tag); public [C#] 5 HtmlControl(String\* tag); public: [C++]String) New(ByVal As Public Sub tag [VB] HtmlControl(tag String); function public [JScript] 9 Description 10 of the instance **Initializes** new a 11 System. Web. UI. Html Controls. Html Control class using the specified tag. 12 This constructor is used to create and initialize a new instance of the 13 System.Web.UI.HtmlControls.HtmlControl class using the specified tag. A string that specifies the tag name of the control. 15 Attributes 16 **TrackViewState** 17 18 AttributeCollection Attributes {get;} public [C#] 19 get Attributes(); property AttributeCollection\* [C++]public: 20 AttributeCollection Attributes Public ReadOnly Property As [VB] 21 AttributeCollection; Attributes() [JScript] public function get 22 23 Description 24 25

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a collection of all attribute name and value pairs expressed on a server control tag within the .aspx file.

Use this property to programmatically access the attributes of the HTML server control. All Html server controls store their attributes in the System.Web.UI.Control.ViewState.

ChildControlsCreated

ClientID

Context

Controls

Disabled

**TrackViewState** 

Description

Gets or sets a value indicating whether the HTML server control is disabled.

On the browser, a disabled element or control is read-only, with the following added restrictions: its value is not submitted with the form, the element or control cannot receive focus, and the element or control is skipped when navigating the document by tabbing.

EnableViewState

**Events** 

HasChildViewState

ID

IsTrackingViewState

NamingContainer
Page
Parent
Site
Style
TrackViewState

## Description

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a collection of all cascading style sheet (CSS) properties applied to a specified HTML server control in the .aspx file.

Use this property to programmatically access the style properties of the HTML server control.

TagName

TrackViewState

**TagName** {get;} virtual string [C#] public public: virtual String\* get TagName(); property [C++]Property [VB] Overridable Public ReadOnly **TagName** String TagName() String; [JScript] public function get

## Description

Gets the element name of a tag that contains a **runat=server** attribute and value pair.

Use this property to programmatically determine the element name of the HTML server control. **TemplateSourceDirectory** 3 UniqueID ViewState ViewStateIgnoresCase **TrackViewState** 9 Description 10 Visible 11 CreateControlCollection 12 13 CreateControlCollection(); ControlCollection protected override [C#] 14 CreateControlCollection(); ControlCollection\* [C++]protected: 15 Function CreateControlCollection() Overrides Protected As [VB] 16 ControlCollection 17 function CreateControlCollection() override [JScript] protected 18 ControlCollection; 19 Render 20 21 Render(HtmlTextWriter void writer); [C#] protected override 22 Render(HtmlTextWriter\* writer); [C++]protected: void 23 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter) 24 protected override function Render(writer : HtmlTextWriter); [JScript]

11	
1	
2	Description
3	RenderAttributes
4	
5	[C#] protected virtual void RenderAttributes(HtmlTextWriter writer);
6	[C++] protected: virtual void RenderAttributes(HtmlTextWriter* writer);
7	[VB] Overridable Protected Sub RenderAttributes(ByVal writer As
8	HtmlTextWriter)
9	[JScript] protected function RenderAttributes(writer : HtmlTextWriter);
10	
11	Description
12	RenderBeginTag
13	
14	[C#] protected virtual void RenderBeginTag(HtmlTextWriter writer);
15	[C++] protected: virtual void RenderBeginTag(HtmlTextWriter* writer);
16	[VB] Overridable Protected Sub RenderBeginTag(ByVal writer As
17	HtmlTextWriter)
18	[JScript] protected function RenderBeginTag(writer : HtmlTextWriter);
19	
20	Description
21	IAttributeAccessor.GetAttribute
22	
23	[C#] string IAttributeAccessor.GetAttribute(string name);
24	[C++] String* IAttributeAccessor::GetAttribute(String* name);
25	[VB] Function GetAttribute(ByVal name As String) As String Implements
- 11	

4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	)
21	
22	?
23	;
24	ļ

IAttributeAccessor.GetAttribute
[JScript] function IAttributeAccessor.GetAttribute(name : String) : String;
IAttributeAccessor.SetAttribute
[C#] void IAttributeAccessor.SetAttribute(string name, string value
[C++] void IAttributeAccessor::SetAttribute(String* name, String* value
[VB] Sub SetAttribute(ByVal name As String, ByVal value As String
Implements IAttributeAccessor.SetAttributeAccessor.
[JScript] function IAttributeAccessor.SetAttribute(name : String, value : String);
HtmlForm class (System.Web.UI.HtmlControls)
TrackViewState
Description
Provides programmatic access to the HTML
Top of Form
element on the server.
The System.Web.UI.HtmlControls.HtmlForm control is used as
container for server controls on a Web page. All server controls that post back
the server must be placed between the opening and closing tags of
System.Web.UI.HtmlControls.HtmlForm control.
HtmlForm
Example Syntax:
TrackViewState

HtmlForm(); public [C#] HtmlForm(); public: [C++]New() Public Sub [VB] HtmlForm(); function public [JScript] 5 6 Description of the instance Initializes new a 8  $System. Web. UI. Html Controls. Html Form \ class.$ 9 Use this constructor to create and initialize a new instance of the 10  $System. Web. UI. Html Controls. Html Form \ class.$ 11 Attributes 12 ChildControlsCreated 13 ClientID 14 Context 15 Controls 16 Disabled 17 **EnableViewState** 18 Enctype 19 **TrackViewState** 20 21 22 Description 23 Gets or sets the encoding type browsers use when posting the form's data to 24 the server.

Use this property to specify the encoding type browsers use to post data back to the server.

**Events** 

HasChildViewState

ID

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

InnerHtml

InnerText

IsTrackingViewState

Method

TrackViewState

## Description

Gets or sets a value that indicates how a browser posts form data to the server for processing.

Use this property to specify how the browser sends form data to the server for processing. The two common methods supported by all browsers are **GET** and **POST** .

Name

TrackViewState

Name {get; set;} public virtual string [C#] [C++] public: \_\_property virtual String\* get\_Name();public: \_\_property virtual set Name(String\*); void As String Name Public Property Overridable [VB]

[JScript] public function get Name(): String; public function set Name(String); 2 Description 3 Gets the identifier for the name System. Web. UI. Html Controls. Html Form control. 5 specific Use this property identify to a 6 System.Web.UI.HtmlControls.HtmlForm control. NamingContainer 8 Page 9 Parent 10 Site 11 Style 12 TagName 13 Target 14 TrackViewState 15 16 17 Description 18 Gets or sets the frame or window to render the results of information posted 19 to the server. 20 Use this property to display the results of information posted to the server 21 in another browser window or frame. 22 TemplateSourceDirectory 23 UniqueID 24 ViewState 25

1	ViewStateIgnoresCase
2	Visible
3	OnInit
4	
5	[C#] protected override void OnInit(EventArgs e);
6	[C++] protected: void OnInit(EventArgs* e);
7	[VB] Overrides Protected Sub OnInit(ByVal e As EventArgs)
8	[JScript] protected override function OnInit(e : EventArgs);
9	
10	Description
11	Call RegisterViewStateHandler().
12	Render
13	
14	[C#] protected override void Render(HtmlTextWriter output);
15	[C++] protected: void Render(HtmlTextWriter* output);
16	[VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)
17	[JScript] protected override function Render(output : HtmlTextWriter);
18	
19	Description
20	
21	RenderAttributes
22	
23	[C#] protected override void RenderAttributes(HtmlTextWriter writer);
24	[C++] protected: void RenderAttributes(HtmlTextWriter* writer);
25	[VB] Overrides Protected Sub RenderAttributes(ByVal writer As

1	HtmlTextWriter)
2	[JScript] protected override function RenderAttributes(writer: HtmlTextWriter);
3	
4	Description
5	RenderChildren
6	
7	[C#] protected override void RenderChildren(HtmlTextWriter writer);
8	[C++] protected: void RenderChildren(HtmlTextWriter* writer);
9	[VB] Overrides Protected Sub RenderChildren(ByVal writer As HtmlTextWriter)
10	[JScript] protected override function RenderChildren(writer : HtmlTextWriter);
11	
12	Description
13	HtmlGenericControl class (System.Web.UI.HtmlControls)
14	TrackViewState
15	
16	
17	Description
18	Defines the methods, properties, and events for all HTML server control
19	tags not represented by a specific .NET Framework class.
20	Use this class to represent an HTML server control tag not directly
21	represented by a .NET Framework class, such as ,
22	,
23	, and .
24	HtmlGenericControl
25	Example Syntax:

TrackViewState 2 public HtmlGenericControl(); [C#] [C++]public: HtmlGenericControl(); Public Sub New() [VB] [JScript] public function HtmlGenericControl(); Initializes a new instance of the System.Web.UI.HtmlControls.HtmlGenericControl class. 8 Description 9 Initializes of the instance new 10 a System.Web.UI.HtmlControls.HtmlGenericControl class with default values. 11 The following table shows initial property values for an instance of 12 System.Web.UI.HtmlControls.HtmlGenericControl. 13 **HtmlGenericControl** 14 Example Syntax: 15 **TrackViewState** 16 17 HtmlGenericControl(string [C#] public tag); 18 [C++]public: HtmlGenericControl(String\* tag); 19 [VB] Public Sub New(ByVal As String) tag 20 [JScript] public function HtmlGenericControl(tag String); 22 Description 23 24 25

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes of new instance System. Web. UI. Html Controls. Html Generic Control class with the specified tag. The following table shows initial property values for an instance of System.Web.UI.HtmlControls.HtmlGenericControl . The name of the element for which the instance of this class is created. Attributes ChildControlsCreated ClientID Context Controls Disabled EnableViewState **Events** HasChildViewState

the

InnerHtml

ID

InnerText

IsTrackingViewState

NamingContainer

Page

Parent

Site

Style

TagName

9

10

11

12

13

15

16

17

18

19

20

21

22

23

24

25

Track	V	1e	W	21	tai	

Description

Gets or sets the tag name of an element that contains a runat= "server" attribute.

Use this property to dynamically change the tag name of a generic control on the Web page.

**TemplateSourceDirectory** 

UniqueID

ViewState

ViewStateIgnoresCase

Visible

HtmlImage class (System.Web.UI.HtmlControls)

TrackViewState

## Description

Provides programmatic access for the HTML element on the server.

Use this control to display an image on a Web page. The **System.Web.UI.HtmlControls.HtmlImage** control can be programmatically manipulated to change the image displayed, the image size, and the alignment of the image relative to other page elements.

HtmlImage

Example Syntax:

**TrackViewState** 2 HtmlImage(); public [C#] [C++]public: HtmlImage(); Public Sub New() [VB] [JScript] public function HtmlImage(); 7 Description Initializes of the instance a new 9 System.Web.UI.HtmlControls.HtmlImage class. 10 Use this constructor to create and initialize a new instance of the 11 System.Web.UI.HtmlControls.HtmlImage class. Align 13 TrackViewState 14 15 public string Align {get; set;} [C#] 16 [C++] public: property String\* get Align();public: property void set Align(String\*); 18 Public Align [VB] Property As String 19 [JScript] public function get Align(): String; public function set Align(String); 21 Description 22 Gets or sets the alignment of the image relative to other Web page 23 elements. 25

Use this property to specify the alignment of the image with respect to other elements on the Web page.

Alt

TrackViewState

public

property

3

10

11

12

13

15

16

17

18

19

20

3

[C++] public:

[C#]

set\_Alt(String\*);

[VB] Public

Property

string

String\*

Alt

Alt

get Alt();public:

As

{get;

property

String

set;}

void

[JScript] public function get Alt() : String; public function set Alt(String);

Description

Gets or sets the alternative caption the browser displays if an image is unavailable or currently downloading and not yet finished.

Use this property to specify the caption displayed when the image specified by the **System.Web.UI.HtmlControls.HtmlImage.Src** property is unavailable. On newer browsers, this caption also appears as a ToolTip.

Attributes

Border

**TrackViewState** 

21 22

24

25

23 Description

Gets or sets the width of a frame for an image.

3

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Use this property to create a frame with the specified width (in pixels) for an image.

ChildControlsCreated

ClientID

Context

Controls

Disabled

EnableViewState

**Events** 

HasChildViewState

Height

**TrackViewState** 

# Description

Gets or sets the height of the image.

The System.Web.UI.HtmlControls.HtmlImage.Height and System.Web.UI.HtmlControls.HtmlImage.Width properties can be used two ways. You can use the System.Web.UI.HtmlControls.HtmlImage.Height and System.Web.UI.HtmlControls.HtmlImage.Width properties to send image size specifications to the browser. This displays the Web page faster because the browser does not need to recalculate the positions of elements on the page when the image loads.

ID

**IsTrackingViewState** 

STATE AND THE STATE OF THE STAT

2

3

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

NamingContainer
Page
Parent
Site

TrackViewState

### Description

Src

Gets or sets the source of the image file to display.

Use this property to specify the path to the image file to display. If the image file is in the same directory as the Web page source file that uses it, you can just specify the file name. Otherwise, you must also include the path to the file. The path can be absolute or relative to the directory that contains the Web page source file.

Style

TagName

TemplateSourceDirectory

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

**TrackViewState** 

lee@hayes plc 509+324+9256 1501 MS1-863US.APP

#### Description

2

3

4

5

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the width of the image.

The System.Web.UI.HtmlControls.HtmlImage.Height and System.Web.UI.HtmlControls.HtmlImage.Width properties can be used two ways. You can use the System.Web.UI.HtmlControls.HtmlImage.Height and System.Web.UI.HtmlControls.HtmlImage.Width properties to send the image size specifications to the browser. This displays the Web page faster because the browser does not need to recalculate the positions of elements on the page when the image loads.

#### RenderAttributes

override RenderAttributes(HtmlTextWriter void [C#] protected writer); [C++]protected: void RenderAttributes(HtmlTextWriter\* writer); [VB] Overrides Protected Sub RenderAttributes(ByVal writer As HtmlTextWriter)

[JScript] protected override function RenderAttributes(writer: HtmlTextWriter);

#### Description

HtmlInputButton class (System.Web.UI.HtmlControls)

TrackViewState

### Description

1    1	Al	lows programn	natic ac	cess to the HTM	AL, <u>Submit</u> , a	$\operatorname{nd}$ Reset el	ements on
2	the server.						
3	Use this class to create button controls on a Web page.						
4	Ht	mlInputButton					
5	Ех	ample Syntax:					
6	TrackViewState						
7							
8	[C#]		ŗ	oublic		HtmlInpu	itButton();
9	[C++]		1	public:		HtmlInpu	itButton();
10	[VB]		Public	;	Sub		New()
11	[JScript]	public function	on Htm	lInputButton();	Initializes a	new instar	nce of an
12	System.V	Veb.UI.HtmlC	Controls	s.HtmlInputBu	tton		class.
13							
14	Descripti	on					
15	In	itializes	a	new	instance	of	an
16	System.V	Veb.UI.HtmlC	Controls	s.HtmlInputBu	tton class usin	ng default v	alues.
17	The following table shows the initial property value for an instance of						nstance of
18	System.Web.UI.HtmlControls.HtmlInputButton .						
19	HtmlInputButton						
20	Example Syntax:						
21	TrackViewState						
22							
23	[C#]	[C#] public HtmlInputButton(string type);					type);
24	[C++]	publi		_	utButton(Stri	_	type);
25	[VB]	Public	Sub	New(ByVa	ıl type	As	String)

24

25.

Context

Controls

Disabled

[JScript] public function HtmlInputButton(type 2 Description Initializes instance a new System.Web.UI.HtmlControls.HtmlInputButton class using the button type. 6 The following table shows the initial property value for an instance of 7 System. Web. UI. Html Controls. Html Input Button. The input button type. 8 Attributes 9 CausesValidation 10 **TrackViewState** 11 12 13 Description 14 Gets or sets a value indicating whether validation is performed when the 15 System. Web. UI. Html Controls. Html Input Button control is clicked. 16 By default, validation performed page is 17 System.Web.UI.HtmlControls.HtmlInputButton control is clicked. 18 validation determines whether the input controls associated with a validation 19 control on the page all pass the validation rules specified by the validation control. 20 ChildControlsCreated 21 ClientID 22

String);

an

specified

of

when

an

EnableViewState **Events** 2 HasChildViewState 3 ID IsTrackingViewState Name NamingContainer Page Parent 9 Site 10 Style 11 TagName TemplateSourceDirectory 13 Type 14 UniqueID 15 Value 16 ViewState 17 ViewStateIgnoresCase 18 Visible 19 TrackViewState 20 21 22

Description

23

24

Occurs when an **System.Web.UI.HtmlControls.HtmlInputButton** control is clicked on the Web page.

This event is raised when an System.Web.UI.HtmlControls.HtmlInputButton control is clicked. 2 OnServerClick 3 protected virtual void OnServerClick(EventArgs [C#] e); [C++]void OnServerClick(EventArgs\* protected: virtual e); [VB] Overridable Protected Sub OnServerClick(ByVal e As EventArgs) OnServerClick(e [JScript] protected function EventArgs); 9 Description 10 Raises the System. Web. UI. Html Controls. Html Input Button. Server Click 11 event. This allows you to handle the event directly. The System.Web.UI.HtmlControls.HtmlInputButton.ServerClick event 13 is raised when an System. Web. UI. Html Controls. Html Input Button control is 14 clicked. A System. EventArgs that contains the event data. 15 RenderAttributes 16 17 [C#] protected override void RenderAttributes(HtmlTextWriter writer); 18 [C++]protected: void RenderAttributes(HtmlTextWriter\* writer); Overrides [VB] Protected Sub RenderAttributes(ByVal writer As 20 HtmlTextWriter) 21 [JScript] protected override function RenderAttributes(writer: HtmlTextWriter); 23 Description 24 IPostBackEventHandler.RaisePostBackEvent 25

[C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument); IPostBackEventHandler::RaisePostBackEvent(String\* [C++]void eventArgument); [VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements IPostBackEventHandler.RaisePostBackEvent [JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument : String); HtmlInputCheckBox class (System.Web.UI.HtmlControls) 9 **TrackViewState** 10 11 12 Description 13 Allows programmatic access to the HTML element on the server. 14 Use the System. Web. UI. Html Controls. Html Input Check Box control to 15 allow the user to select a **true** or **false** state. 16 **HtmlInputCheckBox** 17 Example Syntax: 18 **TrackViewState** 19 20 HtmlInputCheckBox(); [C#] public 21 [C++]public: HtmlInputCheckBox(); [VB] Public Sub New() 23 [JScript] public function HtmlInputCheckBox(); 25

Description

2

3

5

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of an

System.Web.UI.HtmlControls.HtmlInputCheckBox class.

Use this constructor to create and initialize a new instance of the System. Web.UI. HtmlControls. HtmlInputCheckBox class.

Attributes

Checked

**TrackViewState** 

Description

Gets or sets a value indicating whether the System. Web.UI. Html Controls. Html Input Check Box is checked.

Use this property to determine whether the System.Web.UI.HtmlControls.HtmlInputCheckBox control is checked. This property can also be used to programmatically set the state of the System.Web.UI.HtmlControls.HtmlInputCheckBox control.

ChildControlsCreated

ClientID

Context

Controls

Disabled

EnableViewState

**Events** 

HasChildViewState ID

IsTrackingViewState

Name

NamingContainer

Page

Parent

Site

Style

TagName

TemplateSourceDirectory

Type

UniqueID

Value

ViewState

ViewStateIgnoresCase

Visible

TrackViewState

Description

22

23

24

25

Occurs when the Web page is submitted to the server and the  $System. Web. UI. Html Controls. Html Input Check Box\ control\ changes\ state\ from$ the previous post.

2

3

4

5

7

8

9

10

12

13

14

15

16

17

18

20

21

22

23

24

25

This event is raised when the Web page is submitted to the server and the state of the **System.Web.UI.HtmlControls.HtmlInputCheckBox** control changes state from the previous post.

OnPreRender

OnPreRender(EventArgs e); void protected override [C#] e); OnPreRender(EventArgs\* void [C++]protected: EventArgs) OnPreRender(ByVal e As Sub Protected Overrides [VB] EventArgs); function OnPreRender(e override protected [JScript]

Description

OnServerChange

e); OnServerChange(EventArgs protected virtual void [C#] OnServerChange(EventArgs\* e); void virtual protected: [C++][VB] Overridable Protected Sub OnServerChange(ByVal e As EventArgs) OnServerChange(e EventArgs); function protected [JScript]

Description

Raises

**System.Web.UI.HtmlControls.HtmlInputCheckBox.ServerChange** event. This method allows you to handle the event directly.

This event is raised when the Web page is submitted to the server and the state of the **System.Web.UI.HtmlControls.HtmlInputCheckBox** control changes state from the previous post. This method allows you to override the base

implementation and provide a custom handler for the event. A System. EventArgs 1 that contains event information. 2 IPostBackDataHandler.LoadPostData 3 4 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey, 5 NameValueCollection postCollection); IPostBackDataHandler::LoadPostData(String\* postDataKey, [C++]bool NameValueCollection\* postCollection); 8 LoadPostData(ByVal postDataKey String. ByVal Function As [VB] 9 postCollection As NameValueCollection) As Boolean Implements 10 IPostBackDataHandler.LoadPostData 11 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String, 12 postCollection: NameValueCollection): Boolean; 13 IPostBackDataHandler.RaisePostDataChangedEvent 14 15 void IPostBackDataHandler.RaisePostDataChangedEvent(); [C#] 16 void IPostBackDataHandler::RaisePostDataChangedEvent(); [C++]17 Sub RaisePostDataChangedEvent() **Implements** [VB] 18 IPostBackDataHandler.RaisePostDataChangedEvent [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent(); 20 HtmlInputControl class (System.Web.UI.HtmlControls) 21 **TrackViewState** 22 23 24 Description

2

3

4

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Serves as the abstract base class that defines the methods, properties, and events common to all HTML input controls, such as the elements. The System. Web. UI. Html Controls. Html Input Control class cannot be instantiated directly. Instead, this class is inherited by other classes, such as the System.Web.UI.HtmlControls.HtmlInputText System.Web.UI.HtmlControls.HtmlInputButton System. Web. UI. Html Controls. Html Input Radio ButtonSystem.Web.UI.HtmlControls.HtmlInputCheckBox System.Web.UI.HtmlControls.HtmlInputImage and System.Web.UI.HtmlControls.HtmlInputHidden System. Web. UI. Html Controls. Html Input File classes, to provide common basic functionality. HtmlInputControl Example Syntax: **TrackViewState** HtmlInputControl(string type); public [C#] HtmlInputControl(String\* type); public: [C++]String) New(ByVal As [VB] Public Sub type String); HtmlInputControl(type public function [JScript] Description

Initializes a new instance of System. Web. UI. Html Controls. Html Input Control class.

the

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

The following table shows initial property values for an instance of System.Web.UI.HtmlControls.HtmlInputControl. The type of input control.

Attributes

ChildControlsCreated

ClientID

Context

Controls

Disabled

EnableViewState

**Events** 

HasChildViewState

ID

IsTrackingViewState

Name

**TrackViewState** 

## Description

Gets or sets the unique identifier name for the System.Web.UI.HtmlControls.HtmlInputControl.

Use the **System.Web.UI.HtmlControls.HtmlInputControl.Name**property to determine the unique identifier name for an **System.Web.UI.HtmlControls.HtmlInputControl**. In this implementation, the **get** accessor returns the value of the **System.Web.UI.Control.UniqueID**property. However, the **set** accessor does not assign a value to this property.

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

NamingContainer Page Parent Site Style TagName TemplateSourceDirectory Type TrackViewState Description Gets the type of an  ${\bf System. Web. UI. Html Controls. Html Input Control}$  . of type an the this property get Use  $System. Web. UI. Html Controls. Html Input Control\ .$ UniqueID Value TrackViewState Description of contents an the sets Gets or  $System. Web. UI. Html Controls. Html Input Control\ .$ of determine the contents an Use this property to System.Web.UI.HtmlControls.HtmlInputControl . This property can also be

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

HtmlInputFile

Example Syntax:

of the contents an programmatically set used to System.Web.UI.HtmlControls.HtmlInputControl. ViewState ViewStateIgnoresCase Visible RenderAttributes RenderAttributes(HtmlTextWriter writer); override void [C#] protected writer); RenderAttributes(HtmlTextWriter\* void [C++]protected: RenderAttributes(ByVal writer As [VB] **Overrides** Protected Sub HtmlTextWriter) [JScript] protected override function RenderAttributes(writer: HtmlTextWriter); Description HtmlInputFile class (System.Web.UI.HtmlControls) TrackViewState Description Allows programmatic access to the HTML element on the server. Use the System. Web. UI. Html Controls. Html Input File server control to handle uploading binary or text files from a browser client to the server. File upload works with Microsoft Internet Explorer version 3.02 or later.

**TrackViewState** 1 2 HtmlInputFile(); public [C#] 3 public: HtmlInputFile(); [C++]4 Sub New() [VB] Public 5 HtmlInputFile(); public function [JScript] 6 7 Description 8 of the Initializes instance new a 9  $System. Web. UI. Html Controls. Html Input File \ class.$ 10 Use this constructor to create and initialize a new instance of the 11 System. Web. UI. Html Controls. Html Input File class. 12 Accept 13 **TrackViewState** 14 15 set;} public string Accept {get; [C#] 16 get\_Accept();public: [C++]public: property String\* property 17 set Accept(String\*); 18 Accept As String Public Property [VB] 19 [JScript] public function get Accept(): String; public function set Accept(String); 20 21 Description 22 Gets or sets a comma-separated list of MIME encodings used to constrain 23 the file types the user can select. 24

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use this property to specify the file type that can be uploaded to the server. For example, to restrict the selection to images, set this property to "image/\*".

Attributes

ChildControlsCreated

ClientID

Context

Controls

Disabled

EnableViewState

**Events** 

HasChildViewState

ID

IsTrackingViewState

MaxLength

TrackViewState

### Description

Gets or sets the maximum length of the file path for the file to upload from the client machine.

Use this property to specify a limit for the number of characters that can be entered for the path to the file to upload.

Name

NamingContainer

Page

Parent

PostedFile 2 **TrackViewState** 3 4 5 Description Gets access to the uploaded file specified by a client. 7 Browser security restrictions prevent this value from being maintained 8 across multiple requests. 9 Site 10 Size 11 TrackViewState 12 13 14 Description 15 Gets or sets the width of the text box in which the file path is entered. 16 Use this property to specify the width of the text box in which to enter the 17 file path. 18 Style 19 TagName 20 TemplateSourceDirectory 21 Type 22 UniqueID 23 Value 24 ViewState 25

ViewStateIgnoresCase 1 Visible 2 IPostBackDataHandler.LoadPostData 3 4 [C#] IPostBackDataHandler.LoadPostData(string postDataKey, bool 5 NameValueCollection postCollection); 6 IPostBackDataHandler::LoadPostData(String\* [C++]bool postDataKey, 7 NameValueCollection\* postCollection); 8 LoadPostData(ByVal [VB] **Function** postDataKey As String, **ByVal** 9 postCollection As NameValueCollection) As Boolean **Implements** 10 IPostBackDataHandler.LoadPostData 11 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String, 12 postCollection: NameValueCollection): Boolean; 13 IPostBackDataHandler.RaisePostDataChangedEvent 14 15 void IPostBackDataHandler.RaisePostDataChangedEvent(); [C#] 16 [C++]void IPostBackDataHandler::RaisePostDataChangedEvent(); 17 [VB] Sub RaisePostDataChangedEvent() **Implements** 18 IPostBackDataHandler.RaisePostDataChangedEvent 19 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent(); 20 HtmlInputHidden class (System.Web.UI.HtmlControls) 21 TrackViewState 22 23 24 Description

Allows programmatic access to the HTML element on the server. the System. Web. UI. Html Controls. Html Input Hidden You can use 2 control to embed non-visible information within a Top of Form element. This information is sent when the Web page is posted back to the server. HtmlInputHidden Example Syntax: 8 **TrackViewState** 9 10 public HtmlInputHidden(); [C#] 11 [C++]public: HtmlInputHidden(); Sub [VB] Public New() 13 public HtmlInputHidden(); [JScript] function 14 15 Description 16 Initializes of the instance a new 17 System.Web.UI.HtmlControls.HtmlInputHidden class. 18 Attributes 19 ChildControlsCreated 20 ClientID 21 Context 22 Controls 23 Disabled 24 **EnableViewState** 25

**Events** HasChildViewState 2 ID 3 IsTrackingViewState Name 5 NamingContainer 6 Page 7 Parent 8 Site 9 Style 10 TagName 11 TemplateSourceDirectory 12 Type 13 UniqueID 14 Value 15 ViewState 16 ViewStateIgnoresCase 17 Visible 18 TrackViewState 19 20 21 Description 22 Occurs when 23 24

the

System.Web.UI.HtmlControls.HtmlInputControl.Value property is changed on

the server.

25

1	The System.Web.UI.HtmlControls.HtmlInputHidden.ServerChange
2	event is raised when the
3	System.Web.UI.HtmlControls.HtmlInputControl.Value property is changed on
4	the server.
5	OnPreRender
6	
7	[C#] protected override void OnPreRender(EventArgs e)
8	[C++] protected: void OnPreRender(EventArgs* e)
9	[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs
10	[JScript] protected override function OnPreRender(e : EventArgs)
11	
12	Description
13	OnServerChange
14	
15	[C#] protected virtual void OnServerChange(EventArgs e)
16	[C++] protected: virtual void OnServerChange(EventArgs* e)
17	[VB] Overridable Protected Sub OnServerChange(ByVal e As EventArgs)
18	[JScript] protected function OnServerChange(e : EventArgs)
19	
20	Description
21	Raises
22	System.Web.UI.HtmlControls.HtmlInputHidden.ServerChange event.
23	The System.Web.UI.HtmlControls.HtmlInputHidden.ServerChange
24	event is raised when the
25	

System.Web.UI.HtmlControls.HtmlInputControl.Value property is changed on the server. A **System.EventArgs** that contains event data. 2 IPostBackDataHandler.LoadPostData 3 4 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey, NameValueCollection postCollection); 6 [C++]bool IPostBackDataHandler::LoadPostData(String\* postDataKey, NameValueCollection\* postCollection); 8 [VB] **Function** LoadPostData(ByVal postDataKey As String, ByVal 9 postCollection As NameValueCollection) As Boolean **Implements** 10 IPostBackDataHandler.LoadPostData 11 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String, 12 postCollection: NameValueCollection): Boolean; 13 IPostBackDataHandler.RaisePostDataChangedEvent 14 15 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent(): 16 [C++]void IPostBackDataHandler::RaisePostDataChangedEvent(); 17 [VB] Sub RaisePostDataChangedEvent() **Implements** 18 IPostBackDataHandler.RaisePostDataChangedEvent 19 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent(); 20 HtmlInputImage class (System.Web.UI.HtmlControls) 21 **TrackViewState** 22 23 24 Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Allows programmatic access to the HTML element on the server.

Use the **System.Web.UI.HtmlControls.HtmlInputImage** control to create a button that displays an image. You can programmatically control the action associated with the **System.Web.UI.HtmlControls.HtmlInputImage** control by providing an event handler for the

 ${\bf System. Web. UI. Html Controls. Html Input Image. Server Click\ event.}$ 

HtmlInputImage

Example Syntax:

TrackViewState

[C#] public HtmlInputImage();

[C++] public: HtmlInputImage();

[VB] Public Sub New()

[JScript] public function HtmlInputImage();

Description

Initializes a new instance of the

System. Web. UI. Html Controls. Html Input I mage class.

Align

TrackViewState

[C#] public string Align {get; set;}
[C++] public: \_\_property String\* get\_Align();public: \_\_property void
set\_Align(String\*);

[VB] Public Property Align As String

[JScript] public function get Align() : String; public function set Align(String);

Description

Gets or sets the alignment of the

Gets or sets the alignment of the System.Web.UI.HtmlControls.HtmlInputImage control in relation to other elements on the Web page.

Use the **System.Web.UI.HtmlControls.HtmlInputImage.Align** property to specify the alignment of the **System.Web.UI.HtmlControls.HtmlInputImage** control in relation to the other elements on the Web page. The following table lists the possible values for this property.

Alt

TrackViewState

[C#] public string Alt {get; set;}
[C++] public: \_\_property String\* get\_Alt();public: \_\_property void
set\_Alt(String\*);

[VB] Public Property Alt As String [JScript] public function get Alt() : String; public function set Alt(String);

Description

Gets or sets the alternative text that the browser displays if the image is unavailable or has not been downloaded.

Use the System.Web.UI.HtmlControls.HtmlInputImage.Alt property to specify the text to display when the specified image is not available or has not

3

5

6

7

8

9

10

11

13

14

15

16

17

18

19

20

21

22

23

24

25

been downloaded. You can also use this property to programmatically determine the specified alternative text. Attributes Border TrackViewState Description Gets or sets the border width for the System.Web.UI.HtmlControls.HtmlInputImage control. Use the System.Web.UI.HtmlControls.HtmlInputImage.Border the property to specify border width for the System. Web. UI. Html Controls. Html Input I mage control. CausesValidation **TrackViewState** [C#] public bool CausesValidation {get; set;} [C++] public: \_\_property bool get\_CausesValidation();public: \_\_property void set\_CausesValidation(bool); Public [VB] Property CausesValidation As Boolean [JScript] public function get CausesValidation(): Boolean; public function set CausesValidation(Boolean); Description

2

3

5

6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Gets or sets a value indicating whether validation is performed when the **System.Web.UI.HtmlControls.HtmlInputImage** control is clicked.

By default, page validation is performed when an System.Web.UI.HtmlControls.HtmlInputImage control is clicked. Page validation determines whether the input controls associated with a validation control on the page all pass the validation rules specified by the validation control.

ChildControlsCreated

ClientID

Context

Controls

Disabled

EnableViewState

**Events** 

HasChildViewState

ID

IsTrackingViewState

Name

NamingContainer

Page

Parent

Site

Src

**TrackViewState** 

2425

Description

1

2

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the location of the image file.

Use the System. Web. UI. Html Controls. Html Input Image. Src property to specify the location of the image to display in the System. Web. UI. Html Controls. Html Input I mage control. If the image is not available, the specified text in the System.Web.UI.HtmlControls.HtmlInputImage.Alt property is displayed.

Style

TagName

**TemplateSourceDirectory** 

Type

UniqueID

Value

ViewState

ViewStateIgnoresCase

Visible

TrackViewState

Description

Occurs the on server when the clicks user an System.Web.UI.HtmlControls.HtmlInputImage control.

14

17

18

19

20

21

22

23

24

25

The System.Web.UI.HtmlControls.HtmlInputImage.ServerClick event 1 raised is when 2 the user clicks an  ${\bf System. Web. UI. Html Controls. Html Input Image\ control.}$ 3 OnPreRender 5 [C#] protected override void 6 OnPreRender(EventArgs e); [C++]7 protected: void OnPreRender(EventArgs\* e); Overrides Protected Sub OnPreRender(ByVal e As 8 EventArgs) [JScript] protected override 9 function OnPreRender(e EventArgs); 10 Description 11 **OnServerClick** 13 [C#] protected virtual void OnServerClick(ImageClickEventArgs e); [C++]protected: virtual On Server Click (Image Click Event Args \*void 15 e); [VB] Overridable Protected OnServerClick(ByVal Sub As ImageClickEventArgs) [JScript] protected function OnServerClick(e ImageClickEventArgs); : Description Raises the System. Web. UI. Html Controls. Html Input Image. Server Click event. The System.Web.UI.HtmlControls.HtmlInputImage.ServerClick event is raised when the clicks user an

System.Web.UI.HtmlControls.HtmlInputImage control. A 1 System. Web. UI. Image Click Event Args that contains event data. 2 RenderAttributes 3 [C#] protected override RenderAttributes(HtmlTextWriter void writer); 5 [C++]protected: void RenderAttributes(HtmlTextWriter\* writer); 6 [VB] Overrides Protected Sub RenderAttributes(ByVal writer As 7 HtmlTextWriter) 8 [JScript] protected override function RenderAttributes(writer: HtmlTextWriter); 9 10 Description 11 IPostBackDataHandler.LoadPostData 12 13 [C#] bool IPostBackDataHandler.LoadPostData(string 14 postDataKey, NameValueCollection postCollection); 15 [C++]bool IPostBackDataHandler::LoadPostData(String\* postDataKey, 16 NameValueCollection\* postCollection); 17 [VB] Function LoadPostData(ByVal postDataKey As String, **ByVal** 18 postCollection As NameValueCollection) As Boolean **Implements** 19 IPostBackDataHandler.LoadPostData 20 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String. 21 postCollection: NameValueCollection): Boolean; 22 IPostBackDataHandler.RaisePostDataChangedEvent 23 24 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent(); 25

18

19

20

21

22

23

24

25

[C++]void IPostBackDataHandler::RaisePostDataChangedEvent(); [VB] Sub RaisePostDataChangedEvent() 2 **Implements** IPostBackDataHandler. RaisePostDataChangedEvent3 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent(); 4 IPostBackEventHandler.RaisePostBackEvent 5 6 [C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument); 7 [C++]void IPostBackEventHandler::RaisePostBackEvent(String\* 8 eventArgument); 9 [VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements IPostBackEventHandler.RaisePostBackEvent 11 [JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument : 12 String); 13 HtmlInputRadioButton class (System.Web.UI.HtmlControls) 14 **TrackViewState** 15 16

# Description

Allows programmatic access to the HTML element on the server.

Use the **System.Web.UI.HtmlControls.HtmlInputRadioButton** control to create a radio button on a Web page. The **System.Web.UI.HtmlControls.HtmlInputRadioButton** control does not provide built-in functionality to display a caption for the radio button. To create a caption, use literal text in the Web page at the desired location. This allows you to control where the caption is displayed relative to the radio button. For example, if you

want to display the caption on the right side of the radio button, declare an System.Web.UI.HtmlControls.HtmlInputRadioButton control followed by the caption text. 3 HtmlInputRadioButton Example Syntax: **TrackViewState** [C#] public HtmlInputRadioButton(); [C++]public: HtmlInputRadioButton(); [VB] Public Sub New() 10 [JScript] function public HtmlInputRadioButton(); 11 12 Description 13 Initializes instance of the a new 14 System. Web. UI. Html Controls. Html Input Radio Button class. 15 Use this constructor to create and initialize a new instance of the 16 System.Web.UI.HtmlControls.HtmlInputRadioButton class. 17 Attributes 18 Checked 19 **TrackViewState** 20 21 22 Description 23 Gets value indicating whether the or sets 24 System.Web.UI.HtmlControls.HtmlInputRadioButton control is selected.

Use System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked property to determine whether the System. Web. UI. Html Controls. Html Input Radio Button control is selected. You can also use this property to programmatically specify control is the whether System. Web. UI. Html Controls. Html Input Radio Button controls, you need to 6 each iterate through 7 System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked property of 8 each control individually. 9 ChildControlsCreated 10 ClientID 11 Context 12 Controls 13 Disabled 14 EnableViewState 15 **Events** 16 HasChildViewState 17 ID 18 **IsTrackingViewState** 19 Name 20 **TrackViewState** 21 22 23 Description 24

the

of

the

group

test

selected.

If

control

you

have

and

25

4

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets the name of the group that the current instance of the System. Web. UI. Html Controls. Html Input Radio Button class is associated with.

Group multiple System. Web. UI. Html Controls. Html Input Radio Button for the value specifying common a controls together by System.Web.UI.HtmlControls.HtmlInputControl.Name property of each radio When include in the group. want button you System.Web.UI.HtmlControls.HtmlInputRadioButton controls together, only selected at time. the group can be in one radio button System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked property of the selected control is set to true, while the same property is set to false for all other check boxes in the group.

NamingContainer

Page

Parent

Site

Style

TagName

**TemplateSourceDirectory** 

Type

UniqueID

Value

TrackViewState

Description

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the quantity associated with the System.Web.UI.HtmlControls.HtmlInputRadioButton control.

Use the System. Web. UI. Html Controls. Html Input Radio Button. Value with the quantity associate a property to System. Web. UI. Html Controls. Html Input Radio Button control. The quantity is not restricted to a numeric value and can be any valid string. This is useful when you have multiple radio buttons and need perform a calculation based on the selection. For example, you can have radio buttons that represent different the shipping cost in store the methods. You can shipping System. Web. UI. Html Controls. Html Input Radio Button. Value property. When the user selects the shipping method, you add the appropriate amount to the sales total.

ViewState

ViewStateIgnoresCase

Visible

**TrackViewState** 

Description

Occurs when the value of the System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked property of the System.Web.UI.HtmlControls.HtmlInputRadioButton control changes between posts to the server.

The

System.Web.UI.HtmlControls.HtmlInputRadioButton.ServerChange event is

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

the when raised System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked property of the System. Web. UI. Html Controls. Html Input Radio Button control changes value between posts to the server. This allows you to create a custom event handler that performs a specific set of instructions (such as data validation) when the event is raised. OnPreRender override void OnPreRender(EventArgs e); protected [C#] OnPreRender(EventArgs\* e); protected: void [C++]OnPreRender(ByVal e As EventArgs) Sub Overrides Protected [VB] OnPreRender(e EventArgs); function protected override [JScript] Description OnServerChange OnServerChange(EventArgs e); void [C#] protected virtual void OnServerChange(EventArgs\* e); virtual protected: [C++][VB] Overridable Protected Sub OnServerChange(ByVal e As EventArgs) EventArgs); function OnServerChange(e [JScript] protected Description the Raises System. Web. UI. Html Controls. Html Input Radio Button. Server Changeevent. This allows you to create a custom event handler when the event is raised.

> 17 18

> > 19

2021

23

24

25

The

System.Web.UI.HtmlControls.HtmlInputRadioButton.ServerChange event is raised when the System.Web.UI.HtmlControls.HtmlInputRadioButton.Checked property of the System.Web.UI.HtmlControls.HtmlInputRadioButton control changes values between posts to the server. This allows you to create a custom event handler that performs a specific set of instructions (such as data validation)when the event is raised. A System.EventArgs that contains the event data.

RenderAttributes

RenderAttributes(HtmlTextWriter writer); void override [C#] protected RenderAttributes(HtmlTextWriter\* writer); void [C++]protected: writer As Protected Sub RenderAttributes(ByVal **Overrides** [VB] HtmlTextWriter)

[JScript] protected override function RenderAttributes(writer : HtmlTextWriter);

Description

## IPostBackDataHandler.LoadPostData

IPostBackDataHandler.LoadPostData(string postDataKey, [C#] bool postCollection); NameValueCollection IPostBackDataHandler::LoadPostData(String\* postDataKey, bool [C++]postCollection); NameValueCollection\* String, ByVal postDataKey LoadPostData(ByVal Function [VB]

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Boolean **Implements** postCollection NameValueCollection) As As IPostBackDataHandler.LoadPostData [JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String, postCollection: NameValueCollection): Boolean; IPostBackDataHandler. RaisePostDataChangedEventIPostBackDataHandler.RaisePostDataChangedEvent(); void [C#] IPostBackDataHandler::RaisePostDataChangedEvent(); void [C++]RaisePostDataChangedEvent() Implements Sub [VB] IPostBackDataHandler.RaisePostDataChangedEvent [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent(); HtmlInputText class (System.Web.UI.HtmlControls) **TrackViewState** Description HTML and the Allows programmatic access to elements on the server. Use the System. Web. UI. Html Controls. Html Input Text control to create a single line text box that allows the user to enter text or a password. Use the System. Web. UI. Html Controls. Html Input Text. Max Length property to specify the maximum number of characters that can be entered in the text box. The System.Web.UI.HtmlControls.HtmlInputText.Size property allows you to specify the width of the text box. HtmlInputText

```
Example Syntax:
1
           TrackViewState
2
3
                                                                     HtmlInputText();
   [C#]
                                   public
                                                                     HtmlInputText();
                                    public:
    [C++]
                                                                                New()
                                                        Sub
                             Public
    [VB]
6
    [JScript] public function HtmlInputText(); Initializes a new instance of the
7
                                                                                 class.
    System.Web.UI.HtmlControls.HtmlInputText
8
9
    Description
10
                                                                        of
                                                                                   the
                                                      instance
           Initializes
                              a
                                        new
11
    {\bf System. Web. UI. Html Controls. Html Input Text}\ class\ using\ default\ values.
12
           This constructor creates a text type text box control.
13
           HtmlInputText
14
           Example Syntax:
15
           TrackViewState
16
17
                                             HtmlInputText(string
                                                                                 type);
                        public
    [C#]
18
                                            HtmlInputText(String*
                                                                                 type);
                        public:
    [C++]
19
                                                                                String)
                                        New(ByVal
                                                                      As
                Public
                              Sub
                                                           type
    [VB]
20
                                              HtmlInputText(type
                                                                               String);
                   public
                                function
    [JScript]
21
22
    Description
23
24
25
```

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the System.Web.UI.HtmlControls.HtmlInputText class using the specified input control type.

This constructor allows you to create a specific type of text box control, such as **password**. Only the **password** or **text** type is currently supported. This constructor is designed to allow you to create other text box types if they become available. The type of input control.

Attributes

ChildControlsCreated

ClientID

Context

Controls

Disabled

EnableViewState

**Events** 

HasChildViewState

ID

IsTrackingViewState

MaxLength

TrackViewState

**|**|

Description

Gets or sets the maximum number of characters that can be entered in the text box.

1	Use the System. Web. UI. Html Controls. Html Input Text. Max Length
2	property to specify or determine the maximum number of characters that can be
3	entered in the text box.
4	Name
5	NamingContainer
6	Page
7	Parent
8	Site
9	Size
10	TrackViewState
11	
12	
13	Description
14	Gets or sets the width of the text box.
15	Use the System. Web. UI. Html Controls. Html Input Text. Size property to
16	specify or determine the width of the textbox, in characters.
17	Style
18	TagName
19	TemplateSourceDirectory
20	Туре
21	UniqueID
22	Value
23	TrackViewState

# Description

Gets or sets the contents of the text box.

Use the **System.Web.UI.HtmlControls.HtmlInputText.Value** property to programmatically determine the text entered by the user into the text box. You can also use this property to provide default text for the text box.

ViewState

ViewStateIgnoresCase

Visible

TrackViewState

### Description

Occurs when the **System.Web.UI.HtmlControls.HtmlInputText.Value** property is changed on the server.

The System.Web.UI.HtmlControls.HtmlInputText.ServerChange event is raised when the System.Web.UI.HtmlControls.HtmlInputText.Value property is changed on the server.

#### OnPreRender

[C#]	protected	overri	ae	voia	OnFreRend	er(r	eveni.	Aigs	c)
[C++]	prote	ected:	void		OnPreRender(I	Ever	ntArg	s*	e)
[VB]	Overrides	Protected	Sub	OnPre	Render(ByVal	e	As	EventA	rgs]
[ [Scrin	tl protecte	ed overri	de fii	nction	OnPreRende	r(e	:	EventAr	gs)

ū
W.
n paris n paris
ž
ž
in the second

## Description

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

## OnServerChange

OnServerChange(EventArgs e); void virtual protected [C#] OnServerChange(EventArgs\* e); virtual void [C++]protected: [VB] Overridable Protected Sub OnServerChange(ByVal e As EventArgs) EventArgs); OnServerChange(e function [JScript] protected

# Description

Raises the System. Web. UI. Html Controls. Html Input Text. Server Change event.

 $The \ \ System. Web. UI. Html Controls. Html Input Text. Server Change \ \ event$ System. Web. UI. Html Controls. Html Input Text. Valuethe property is changed on the server. A System. EventArgs that contains event data.

#### RenderAttributes

writer); RenderAttributes(HtmlTextWriter override void protected [C#] RenderAttributes(HtmlTextWriter\* writer); void [C++]protected: writer As Protected Sub RenderAttributes(ByVal [VB] Overrides HtmlTextWriter)

[JScript] protected override function RenderAttributes(writer: HtmlTextWriter);

#### Description

1543 MS1-863US.APP lee@haves plic 509-324-9256

## IPostBackDataHandler.LoadPostData

2

3

5

6

8

9

10

11

13 14

15

17

16

18 19

21

20

22

2324

25

[C#]	bool	IPostBackDataHandler.LoadPostData(string	postDataKey,
NameVa	alueColle	ction	postCollection);

[C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey, NameValueCollection\* postCollection);

[VB] Function LoadPostData(ByVal postDataKey As String, ByVal postCollection As NameValueCollection) As Boolean Implements IPostBackDataHandler.LoadPostData

[JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String, postCollection : NameValueCollection) : Boolean;

IPostBackDataHandler. RaisePostDataChangedEvent

```
[C#] void IPostBackDataHandler.RaisePostDataChangedEvent();
```

[C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

[VB] Sub RaisePostDataChangedEvent() Implements

IPostBackDataHandler. RaisePostDataChangedEvent

[JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

HtmlSelect class (System.Web.UI.HtmlControls)

TrackViewState

Description

Allows programmatic access to the HTML

## System.Web.UI.WebControls

# Description

The System.Web.UI.WebControls namespace is a collection of classes that allow you to create Web server controls on a Web page. Web controls run on the server and include form controls such as buttons and text boxes, as well as special purpose controls such as a calendar. This allows you to programmatically control these elements on a Web page. Web controls are more abstract than HTML controls. Their object model does not necessarily reflect HTML syntax.

AdCreatedEventArgs class (System.Web.UI.WebControls)

### Description

Provides data for the

System.Web.UI.WebControls.AdRotator.AdCreated event of the

System. Web. UI. Web Controls. Ad Rotator control. This class cannot be inherited.

The **System.Web.UI.WebControls.AdRotator.AdCreated** event is raised when the **System.Web.UI.WebControls.AdRotator** displays an advertisement on the page.

Constructors:

AdCreatedEventArgs

Example Syntax:

1	
2	[C#] public AdCreatedEventArgs(IDictionary adProperties);
3	[C++] public: AdCreatedEventArgs(IDictionary* adProperties);
4	[VB] Public Sub New(ByVal adProperties As IDictionary)
5	[JScript] public function AdCreatedEventArgs(adProperties : IDictionary);
6	
7	Description
8	Initializes a new instance of the
9	System.Web.UI.WebControls.AdCreatedEventArgs class.
10	Use this constructor to create and initialize a new instance of the
11	System.Web.UI.WebControls.AdCreatedEventArgs class. A
12	System.Collections.IDictionary containing the advertisement properties from the
13	XML file.
14	Properties:
15	AdProperties
16	
17	[C#] public IDictionary AdProperties {get;}
18	[C++] public:property IDictionary* get_AdProperties();
19	[VB] Public ReadOnly Property AdProperties As IDictionary
20	[JScript] public function get AdProperties(): IDictionary;
21	
22	Description
23	Gets a System.Collections.IDictionary object that contains all the
24	advertisement properties for the currently displayed advertisement.
25	

6

7

10

11

12

13

14

15

16

18

19

20

21

22

23

Use the

System.Web.UI.WebControls.AdCreatedEventArgs.AdProperties property to get a System.Collections.IDictionary object that contains the advertisement properties for the currently displayed advertisement. The keys and values of the System.Collections.IDictionary object are of type System.String.

AlternateText

[C#] public string AlternateText {get; set;}

[C++] public: \_\_property String\* get\_AlternateText();public: \_\_property void set\_AlternateText(String\*);

[VB] Public Property AlternateText As String

[JScript] public function get AlternateText() : String;public function set AlternateText(String);

### Description

Gets or sets the alternate text displayed in the

**System.Web.UI.WebControls.AdRotator** control when the advertisement image is unavailable. Browsers that support the ToolTips feature display this text as a ToolTip for the advertisement.

Use the

System.Web.UI.WebControls.AdCreatedEventArgs.AlternateText property to specify the text to display if the image specified in the

**System.Web.UI.WebControls.AdCreatedEventArgs.ImageUrl** property is not available. In browsers that support the ToolTips feature, this text also displays as a ToolTip for the advertisement.

24 25

ImageU	rl
mugoo	_ ,

[C#] public string ImageUrl {get; set;}

[C++] public: \_\_property String\* get\_ImageUrl();public: \_\_property void set\_ImageUrl(String\*);

[VB] Public Property ImageUrl As String

[JScript] public function get ImageUrl(): String;public function set ImageUrl(String);

Description

Gets or sets the URL of an image to display in the System. Web.UI. WebControls. AdRotator control.

Use the **System.Web.UI.WebControls.AdCreatedEventArgs.ImageUrl** property to specify the URL of an image to display for an advertisement in the **System.Web.UI.WebControls.AdRotator** control. You can use a relative or an absolute URL. A relative URL relates the location of the image to the location of the Web page without specifying a complete path on the server. The path is relative to the location of the Web page. This makes it easier to move the entire site to another directory on the server without updating the path to the image in code. An absolute URL provides the complete path, so moving the site to another directory requires updating the code.

NavigateUrl

[C#] public string NavigateUrl {get; set;}

[C++] public: \_\_property String\* get\_NavigateUrl();public: \_\_property void

set_NavigateUrl(String*);
[VB] Public Property NavigateUrl As String
[JScript] public function get NavigateUrl(): String; public function set
NavigateUrl(String);
Description
Gets or sets the Web page to display when the
System.Web.UI.WebControls.AdRotator control is clicked.
Use the
System.Web.UI.WebControls.AdCreatedEventArgs.NavigateUrl property to
specify the Web page to display when the
System.Web.UI.WebControls.AdRotator control is clicked.
Methods:
AdCreatedEventHandler delegate (System.Web.UI.WebControls)
ToString
Description
Represents the method that will handle the
System.Web.UI.WebControls.AdRotator.AdCreated event of an
System.Web.UI.WebControls.AdRotator control. The source of the event. An
System.Web.UI.WebControls.AdCreatedEventArgs that contains the event
data.

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

The System.Web.UI.WebControls.AdRotator.AdCreated event is raised when the System. Web.UI. WebControls. AdRotator displays an advertisement on the page. AdRotator class (System.Web.UI.WebControls) **ToString** Description Displays an advertisement banner on a Web page. Use the System.Web.UI.WebControls.AdRotator control to display a randomly selected advertisement banner on the Web page. The displayed advertisement can change whenever the page refreshes. AdRotator Example Syntax: **ToString** 

[C#] public AdRotator();

[C++] public: AdRotator();

[VB] Public Sub New()

[JScript] public function AdRotator();

Description

Initializes a new instance of the System.Web.UI.WebControls.AdRotator class.

23

24

25

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.AdRotator** class.

AccessKey

AdvertisementFile

**ToString** 

Description

Gets or sets the path to an XML file that contains advertisement information.

Use the System.Web.UI.WebControls.AdRotator.AdvertisementFile property to specify the location of an XML file that contains advertisement information. The XML file must reside within the same web site. For deployment and security purposes, you should place the file in the same web application. It is only possible to access an XML file that is located in a different application on the same site if the application has sufficient trust.

Attributes

BackColor

BorderColor

BorderStyle

**BorderWidth** 

ChildControlsCreated

ClientID

Context

Controls

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

1	ControlStyle
2	ControlStyleCreated
3	CssClass
4	Enabled
5	EnableViewState
6	Events
7	Font
8	ToString
9	

# Description

Font property. Has no effect on this control, so hide it.

ForeColor

HasChildViewState

Height

ID

IsTrackingViewState

KeywordFilter

**ToString** 

Description

Gets or sets a category keyword to filter for specific types of advertisements in the XML advertisement file.

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Each advertisement in the XML advertisement file can be assigned a category keyword. Use the

System.Web.UI.WebControls.AdRotator.KeywordFilter property to filter the advertisements for the specified keyword. Only advertisements containing the keyword will be selected for the System.Web.UI.WebControls.AdRotator control. The System.Web.UI.WebControls.AdRotator.KeywordFilter property can be programmatically set to match the profile of the user.

NamingContainer

Page

Parent

Site

Style

TabIndex

TagKey

**TagName** 

**Target** 

**ToString** 

Description

Gets or sets the name of the browser window or frame that displays the contents of the Web page linked to when the

System.Web.UI.WebControls.AdRotator control is clicked.

Use the **System.Web.UI.WebControls.AdRotator.Target** property to specify the target window or frame that displays the contents of the Web page

linked to when the System. Web.UI. WebControls. AdRotator control is clicked.
Values must begin with a letter in the range of a to z (case insensitive), except for
the following special values, which begin with an underscore: _blank Renders the
content in a new, unframed window.
TemplateSourceDirectory
ToolTip
UniqueID
ViewState
ViewStateIgnoresCase
Visible

Description

Width

**ToString** 

Occurs once per round trip to the server after the creation of the control, but before the page is rendered.

This event is raised once per round trip to the server after the creation of the control, but before the page is rendered.

CreateControlCollection

[C#] protected override ControlCollection CreateControlCollection();[C++] protected: ControlCollection\* CreateControlCollection();[VB] Overrides Protected Function CreateControlCollection() AsControlCollection

ControlCollection; 2 3 Description 5 **OnAdCreated** 6 7 [C#] protected virtual void OnAdCreated(AdCreatedEventArgs e); 8 [C++] protected: virtual void OnAdCreated(AdCreatedEventArgs\* e); 9 [VB] Overridable Protected Sub OnAdCreated(ByVal e As AdCreatedEventArgs) 10 [JScript] protected function OnAdCreated(e : AdCreatedEventArgs); 11 12 Description 13 Raises the System. Web. UI. Web Controls. AdRotator. AdCreated event 14 for the System. Web. UI. Web Controls. AdRotator control. 15 The System.Web.UI.WebControls.AdRotator.AdCreated event is raised 16 on the server after the System. Web.UI. WebControls. AdRotator control is 17 created, but before the page is rendered. If the 18 System.Web.UI.WebControls.AdRotator.AdvertisementFile property is set, 19 the System.Web.UI.WebControls.AdRotator.AdCreated event is raised after an 20 advertisement has been selected by the 21 System.Web.UI.WebControls.AdRotator control. An 22 System.Web.UI.WebControls.AdCreatedEventArgs that contains event data. 23 OnPreRender 24 25

[JScript] protected override function CreateControlCollection():

1	
2	[C#] protected override void OnPreRender(EventArgs e);
3	[C++] protected: void OnPreRender(EventArgs* e);
4	[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)
5	[JScript] protected override function OnPreRender(e: EventArgs);
6	
7	Description
8	Gets the advertisement information for rendering by looking up the file data
9	and/or calling the user event.
10	Randomly picks an advertisement based on the impresssion (weight) value
11	of the advertisement.
12	Render
13	
14	[C#] protected override void Render(HtmlTextWriter writer);
15	[C++] protected: void Render(HtmlTextWriter* writer);
16	[VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)
17	[JScript] protected override function Render(writer : HtmlTextWriter);
18	
19	Description
20	Displays the System.Web.UI.WebControls.AdRotator on the client.
21	Displays an image obtained from the ad chosen in
22	System.Web.UI.WebControls.AdRotator.OnPreRender(System.EventArgs).
23	A System.Web.UI.HtmlTextWriter that contains the output stream to render on
24	the client.
25	BaseCompareValidator class (System.Web.UI.WebControls)

	1	TrackViewState
	2	
	3	
		Description
	4	Serves as the abstract base class for validation controls that perform
	5	
	6	comparisons.
	7	The System.Web.UI.WebControls.BaseCompareValidator class is
	8	inherited by validation controls that compare values, such as the
	9	System.Web.UI.WebControls.CompareValidator and
	10	System.Web.UI.WebControls.RangeValidator controls.
	11	BaseCompareValidator
	12	Example Syntax:
	13	TrackViewState
	14	
	15	[C#] protected BaseCompareValidator();
	16	[C++] protected: BaseCompareValidator();
	17	[VB] Protected Sub New()
	18	[JScript] protected function BaseCompareValidator();
	19	AccessKey
	20	Attributes
	21	BackColor
	22	BorderColor
	23	BorderStyle
	24	BorderWidth
	25	ChildControlsCreated

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

ClientID Context

Controls

ControlStyle

ControlStyleCreated

ControlToValidate

CssClass

CutoffYear

**TrackViewState** 

# Description

Gets the maximum year that can be represented by a two-digit year.

Use the

 ${\bf System. Web. UI. WebControls. Base Compare Validator. Cutoff Year\ property\ to}$ determine the maximum year that can be represented by a two-digit year in a 100 year range. For example, if this property contains the value 2029, you can represent the years 1930 to 2029 by using a two-digit year. The two-digit year 30 is interpreted as 1930, while 29 is interpreted as 2029. You cannot change the value of this property directly, however you can change the maximum year that can be represented by a two-digit year by setting the

 ${\bf System. Globalization. Calendar. Two Digit Year Max\ property.}$ 

Display

EnableClientScript

Enabled

1	EnableViewState
2	ErrorMessage
3	Events
4	Font
5	ForeColor
6	HasChildViewState
7	Height
8	ID
9	IsTrackingViewState
10	IsValid
11	NamingContainer
12	Page
13	Parent
14	PropertiesValid
15	RenderUplevel
16	Site
17	Style
18	TabIndex
19	TagKey
20	TagName
21	TemplateSourceDirectory
22	Text
23	ToolTip
24	Туре
25	TrackViewState

$\mathbf{r}$		_
$\nu$	escription	ı

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the data type of the values being compared.

Use the **System.Web.UI.WebControls.BaseCompareValidator.Type** property to specify the data type of the two values being compared. Both values are converted to the specified data type before any comparison is performed.

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

[C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As

HtmlTextWriter)

[JScript] protected override function AddAttributesToRender(writer:

HtmlTextWriter);

Description

AddAttributesToRender method AddAttributesToRender method

CanConvert

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

25

[C#] public static bool CanConvert(string text, ValidationDataType type); [C++] public: static bool CanConvert(String\* text, ValidationDataType type); [VB] Public Shared Function CanConvert(ByVal text As String, ByVal type As ValidationDataType) As Boolean [JScript] public static function CanConvert(text : String, type : ValidationDataType): Boolean;

Description

Determines whether the specified string can be converted to the specified data type.

Return Value: true if the specified data string is convertible to the specified data type; otherwise, false.

Use the

System. Web. UI. WebControls. BaseCompareValidator. CanConvert (System. Structure) and the structure of thring, System. Web. UI. Web Controls. Validation Data Type) method to determine whether the specified string can be converted to the specified data type. This method is commonly used to test whether a string can be converted to a compatible data type before performing an operation that depends on that data type. The string to test whether the specified data type conversion is possible. One of the System. Web. UI. Web Controls. Validation Data Type enumeration values.

Compare

[C#] protected static bool Compare(string leftText, string rightText, ValidationCompareOperator op, ValidationDataType type);

[C++] protected: static bool Compare(String\* leftText, String\* rightText,
ValidationCompareOperator op, ValidationDataType type);
[VB] Protected Shared Function Compare(ByVal leftText As String, ByVal
rightText As String, ByVal op As ValidationCompareOperator, ByVal type As
ValidationDataType) As Boolean
[JScript] protected static function Compare(leftText: String, rightText: String, op
: ValidationCompareOperator, type: ValidationDataType): Boolean;

Description

Compares two strings using the specified operator and validation data type.

Return Value: true if the conversion is successful; otherwise false. The string

Return Value: true if the conversion is successful; otherwise false. The string value on the left side of the operator. The string value on the right side of the operator. A System.Web.UI.WebControls.ValidationCompareOperator object that represents the comparison operation to perform. A System.Web.UI.WebControls.ValidationDataType object that represents the data type of the operands.

Convert

[C#] protected static bool Convert(string text, ValidationDataType type, out object value);

[C++] protected: static bool Convert(String\* text, ValidationDataType type, Object\*\* value);

[VB] Protected Shared Function Convert(ByVal text As String, ByVal type As ValidationDataType, ByRef value As Object) As Boolean

[JScript] protected static function Convert(text : String, type :

3

4

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

ValidationDataType, value: Object): Boolean; Description Converts the specified text into an object with specified validation data type. Return Value: true if the conversion is successful; otherwise false. The source text to convert from. A System.Web.UI.WebControls.ValidationDataType object that represents the data type to convert the text into. A System.Object that contains the conversion result. DetermineRenderUplevel [C#] protected override bool DetermineRenderUplevel(); [C++] protected: bool DetermineRenderUplevel(); [VB] Overrides Protected Function DetermineRenderUplevel() As Boolean [JScript] protected override function DetermineRenderUplevel(): Boolean; Description Determines whether the validation control can be rendered for an uplevel browser. Return Value: true if the validation control can be rendered for an uplevel browser; otherwise, false. GetDateElementOrder [C#] protected static string GetDateElementOrder();

[C++] protected: static String\* GetDateElementOrder();

25

[VB] Protected Shared Function GetDateElementOrder() As String [JScript] protected static function GetDateElementOrder() : String;

Description

Return the order of date elements for the current culture Return the order of date elements for the current culture

GetFullYear

[C#] protected static int GetFullYear(int shortYear);

[C++] protected: static int GetFullYear(int shortYear);

[VB] Protected Shared Function GetFullYear(ByVal shortYear As Integer) As Integer

[JScript] protected static function GetFullYear(shortYear: int): int;

Description

Generates the four-digit year representation of the specified two-digit year.

Return Value: The four-digit year representation of the specified two-digit year.

Use the

System. Web. UI. WebControls. BaseCompare Validator. GetFull Year (System. In the control of t

t32) method to generate the four-digit year representation of the specified two-

digit year. The four-digit year representation depends on the value of the

System.Globalization.BaseCompareValidator.CutoffYear property, which

contains the maximum year that can be represented by a two-digit year in a 100

year range. For example, if the

 ${\bf System. Globalization. Base Compare Validator. Cutoff Year}\ property\ contains$ 

the value 2029, the System. Web. UI. WebControls. BaseCompare Validator. GetFull Year (System. In the control of tt32) method returns a year between 1930 to 2029. The two-digit year 30 is interpreted as 1930, while 29 is interpreted as 2029. You can change the maximum year that can be represented by a two-digit year by setting the System.Globalization.Calendar.TwoDigitYearMax property. A two-digit year. 6 BaseDataList class (System.Web.UI.WebControls) 7 Validate 8 9 10 Description 11 Serves as the abstract base class for data list controls, such as the 12  $System. Web. UI. Web Controls. Data List\ and$ 13 System.Web.UI.WebControls.DataGrid controls. This class provides methods 14 and properties that are common to all data list controls. 15 The System.Web.UI.WebControls.BaseDataList class provides the 16 common functionality for all data list controls, such as the 17  $System. Web. UI. Web Controls. Data List\ and$ 18 System.Web.UI.WebControls.DataGrid controls. An instance of this class is not 19 created directly. Instead, data list controls inherit the methods and properties 20 common to all data list controls from this class. 21 BaseDataList 22 Example Syntax: 23 Validate 24

```
[C#] public BaseDataList();
    [C++] public: BaseDataList();
    [VB] Public Sub New()
    [JScript] public function BaseDataList();
6
    Description
           Initializes a new instance of the
8
    System.Web.UI.WebControls.BaseDataList class.
9
           A System. Web. UI. WebControls. BaseDataList object is not created
10
    directly. This constructor is commonly called in the constructor of a derived class
11
    to initialize the properties defined in the
12
    System.Web.UI.WebControls.BaseDataList class.
13
           AccessKey
14
           Attributes
15
           BackColor
16
           BorderColor
17
           BorderStyle
18
           BorderWidth
19
           CellPadding
20
           Validate
21
22
23
    Description
24
```

Gets or sets the amount of space between the contents of a cell and the cell's border.

Use the **System.Web.UI.WebControls.BaseDataList.CellPadding** property to control the spacing between the contents of a cell and the cell's border. The padding amount specified is added to all four sides of a cell. All cells in the same column of the list control share the same cell width. Therefore, if the content of one cell is longer than the content other cells in the same column, the padding amount is applied to the width of longest cell. All other cells in the column will also share the same cell width. The same is true for all cells in the same row. The padding amount is applied to the tallest cell in the row, with all cells in the same row sharing the same cell height. Individual cell sizes cannot be specified.

CellSpacing

Validate

[C#] public virtual int CellSpacing {get; set;}

[C++] public: \_\_property virtual int get\_CellSpacing();public: \_\_property virtual void set CellSpacing(int);

[VB] Overridable Public Property CellSpacing As Integer

[JScript] public function get CellSpacing(): int;public function set

CellSpacing(int);

Description

Gets or sets the amount of space between cells.

Use the System. Web. UI. WebControls. BaseDataList. Cell Spacing 1 property to control the spacing between adjacent cells in the list control. This 2 spacing is applied both vertically and horizontally. 3 Child Controls CreatedClientID 5 Context Controls Validate 8 9 10 Description 12 ControlStyle 13 ControlStyleCreated 14 CssClass 15 DataKeyField 16 Validate 17 18 19 Description 20 Gets or sets the primary key field in the data source referenced by 21 System. Web. UI. WebControls. BaseDataList. Data Source.22 DataKeys 23 Validate 24

[C#] public DataKeyCollection DataKeys {get;}
[C++] public:property DataKeyCollection* get_DataKeys();
[VB] Public ReadOnly Property DataKeys As DataKeyCollection
[JScript] public function get DataKeys(): DataKeyCollection;
Description
Gets a collection that stores the primary key values of each record

Use the **System.Web.UI.WebControls.BaseDataList.DataKeyCollection** to access the primary key values of each record (displayed as a row) in the list control. This property is commonly used to include the primary key field with the list control without displaying it in the control. The collection is filled with the values from the field specified by the

 $System. Web. UI. Web Controls. Base Data List. Data Key Field\ property.$ 

DataKeysArray

(displayed as a row) in the list control.

Validate

[C#] protected ArrayList DataKeysArray {get;}

[C++] protected: \_\_property ArrayList\* get\_DataKeysArray();

[VB] Protected ReadOnly Property DataKeysArray As ArrayList

[JScript] protected function get DataKeysArray(): ArrayList;

Description

Gets a list of key fields in the data source.

4	
5	
6	
7	
8	
9	Ì
10	
11	
12	
13	
14	
15	
16	l
17	İ
18	
19	
20	
21	
22	

1	DataMember
2	Validate
3	
4	[C#] public string DataMember {get; set;}
5	[C++] public:property String* get_DataMember();public:property void
6	set_DataMember(String*);
7	[VB] Public Property DataMember As String
8	[JScript] public function get DataMember(): String;public function set
9	DataMember(String);
10	
11	Description
12	Gets or sets the specific data member in a multi member
13	System.Web.UI.WebControls.BaseDataList.DataSource to bind to the list
14	control.
15	Use the System.Web.UI.WebControls.BaseDataList.DataMember
16	property to specify a member from a multi member
17	System.Web.UI.WebControls.BaseDataList.DataSource to bind to the list
18	control. For example, if you had a
19	System.Web.UI.WebControls.BaseDataList.DataSource that contains multiple
20	tables, you can use the
21	System.Web.UI.WebControls.BaseDataList.DataMember property to specify
22	the table to bind to the list control.
23	DataSource
24	Validate

1	
2	[C#] public virtual object DataSource {get; set;}
3	[C++] public:property virtual Object* get_DataSource();public:property
4	virtual void set_DataSource(Object*);
5	[VB] Overridable Public Property DataSource As Object
6	[JScript] public function get DataSource(): Object; public function set
7	DataSource(Object);
8	
9	Description
10	Gets or sets the source to a list of values used to populate the items within
11	the control.
12	Lists derived from ICollection can also be used as a DataSource.
13	Enabled
14	EnableViewState
15	Events
16	Font
17	ForeColor
18	GridLines
19	Validate
20	
21	
22	Description
23	Gets or sets a value that specifies the grid line style.
24	HasChildViewState
25	Height

HorizontalAlign Validate 2 3 4 Description 5 Gets or sets the horizontal alignment of the control within its container. 6 ID Is Tracking View StateNamingContainer Page 10 Parent 11 Site 12 Style 13 TabIndex 14 TagKey 15 TagName 16 TemplateSourceDirectory 17 ToolTip18 UniqueID 19 ViewState 20 ViewStateIgnoresCase 21 Visible 22 Width 23 Validate 24 25

1	
2	
3	Description
4	Occurs when an item on the list is selected.
5	AddParsedSubObject
6	
7	[C#] protected override void AddParsedSubObject(object obj);
8	[C++] protected: void AddParsedSubObject(Object* obj);
9	[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)
10	[JScript] protected override function AddParsedSubObject(obj : Object);
11	
12	Description
13	Overridden. The object to add.
14	CreateChildControls
15	
16	[C#] protected override void CreateChildControls();
17	[C++] protected: void CreateChildControls();
18	[VB] Overrides Protected Sub CreateChildControls()
19	[JScript] protected override function CreateChildControls();
20	
21	Description
22	Creates a child control using the view state.
23	CreateControlHierarchy
24	
25	[C#] protected abstract void CreateControlHierarchy(bool useDataSource)

1	[C++] protected: virtual void CreateControlHierarchy(bool useDataSource) = 0;
2	[VB] MustOverride Protected Sub CreateControlHierarchy(ByVal useDataSource
3	As Boolean)
4	[JScript] protected abstract function CreateControlHierarchy(useDataSource:
5	Boolean);
6	
7	Description
8	Overridden. true to use the
9	System.Web.UI.WebControls.BaseDataList.DataSource; otherwise, false.
10	DataBind
11	
12	[C#] public override void DataBind();
13	[C++] public: void DataBind();
14	[VB] Overrides Public Sub DataBind()
15	[JScript] public override function DataBind();
16	
17	Description
18	Binds the control and all of its child controls to the data source specified by
19	the System.Web.UI.WebControls.BaseDataList.DataSource property.
20	IsBindableType
21	
22	[C#] public static bool IsBindableType(Type type);
23	[C++] public: static bool IsBindableType(Type* type);
24	[VB] Public Shared Function IsBindableType(ByVal type As Type) As Boolean
25	[JScript] public static function IsBindableType(type: Type): Boolean;

Description

Determines whether the specified data type is bindable to a list control that derives from the **System.Web.UI.WebControls.BaseDataList** class.

Return Value: true if the specified data type is bindable to a list control that derives from the System.Web.UI.WebControls.BaseDataList class; otherwise, false.

Use the

System.Web.UI.WebControls.BaseDataList.IsBindableType(System.Type)
static method to determine whether the specified data type is bindable to a list
control that inherits from the System.Web.UI.WebControls.BaseDataList class.
The supported data types are System.Boolean, System.Byte, System.SByte,
System.Int16, System.UInt16, System.Int32, System.UInt32, System.Int64,
System.UInt64, System.Char, System.Double, System.Single,
System.DateTime, System.Decimal, and string. A System.Type object that
contains the data type to test.

**OnDataBinding** 

[C#] protected override void OnDataBinding(EventArgs e);

[C++] protected: void OnDataBinding(EventArgs\* e);

[VB] Overrides Protected Sub OnDataBinding(ByVal e As EventArgs)

[JScript] protected override function OnDataBinding(e : EventArgs);

Description

Raises the **DataBinding** event of a 1 System.Web.UI.WebControls.BaseDataList. 2 The event is raised when data is bound to the control. An 3 System. Event Args that contains the event data. 4 OnSelectedIndexChanged 5 6 [C#] protected virtual void OnSelectedIndexChanged(EventArgs e); 7 [C++] protected: virtual void OnSelectedIndexChanged(EventArgs\* e); 8 [VB] Overridable Protected Sub OnSelectedIndexChanged(ByVal e As 9 EventArgs) 10 [JScript] protected function OnSelectedIndexChanged(e: EventArgs); 11 12 Description 13 Raises the 14 System.Web.UI.WebControls.BaseDataList.SelectedIndexChanged event of a 15 System.Web.UI.WebControls.BaseDataList . An System.EventArgs that 16 contains the event data. 17 PrepareControlHierarchy 18 19 [C#] protected abstract void PrepareControlHierarchy(); 20 [C++] protected: virtual void PrepareControlHierarchy() = 0; 21 [VB] MustOverride Protected Sub PrepareControlHierarchy() 22

[JScript] protected abstract function PrepareControlHierarchy();

Description

23

Overridden. Render 2 3 [C#] protected override void Render(HtmlTextWriter writer); [C++] protected: void Render(HtmlTextWriter\* writer); 5 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter) 6 [JScript] protected override function Render(writer: HtmlTextWriter); 8 Description 9 Displays the control on the client. An System.Web.UI.HtmlTextWriter 10 that contains the output stream to render on the client. 11 BaseValidator class (System.Web.UI.WebControls) 12 **TrackViewState** 13 14 15 Description 16 Serves as the abstract base class for validation controls. 17 The System.Web.UI.WebControls.BaseValidator class provides the core 18 implementation for all validation controls. 19 **BaseValidator** 20 Example Syntax: 21 **TrackViewState** 22 23 [C#] protected BaseValidator(); 24 [C++] protected: BaseValidator();

[VB] Protected Sub New() [JScript] protected function BaseValidator(); 2 3 Description Initializes a new instance of the 5  $System. Web. UI. Web Controls. Base Validator \ class.$ 6 This constructor is not called directly. Validation controls that inherit from 7 this class can call this constructor from their own constructors to initialize the base properties. 9 AccessKey 10 Attributes 11 BackColor 12 BorderColor 13 BorderStyle 14 BorderWidth 15 ChildControlsCreated 16 ClientID 17 Context 18 Controls 19 ControlStyle 20 ControlStyleCreated 21 ControlToValidate 22 **TrackViewState** 23 24

Description

Gets or sets the input control to validate.

Use the

**System.Web.UI.WebControls.BaseValidator.ControlToValidate** property to specify the input control to validate. This property must be set to the ID of an input control for all validation controls except for the

System.Web.UI.WebControls.CustomValidator control, which can be left blank. If you do not specify a valid input control, an exception will be thrown when the page is rendered. The ID must refer to a control within the same container as the validation control. It must be in the same page or user control, or it must be in the same template of a templated control.

CssClass

Display

**TrackViewState** 

Description

Gets or sets the display behavior of the error message in a validation control.

Use the **System.Web.UI.WebControls.BaseValidator.Display** property to specify the display behavior of the error message in the validation control. The following table lists the different values that can be used.

EnableClientScript

1	TrackViewState
2	
3	[C#] public bool EnableClientScript {get; set;}
4	[C++] public:property bool get_EnableClientScript();public:property void
5	set_EnableClientScript(bool);
6	[VB] Public Property EnableClientScript As Boolean
7	[JScript] public function get EnableClientScript(): Boolean; public function set
8	EnableClientScript(Boolean);
9	
10	Description
11	Gets or sets a value indicating whether client-side validation is enabled.
12	Use the
13	System.Web.UI.WebControls.BaseValidator.EnableClientScript property to
14	specify whether client-side validation is enabled.
15	Enabled
16	TrackViewState
17	
18	[C#] public override bool Enabled {get; set;}
19	[C++] public:property virtual bool get_Enabled();public:property virtual
20	void set_Enabled(bool);
21	[VB] Overrides Public Property Enabled As Boolean
22	[JScript] public function get Enabled(): Boolean; public function set
23	Enabled(Boolean);
24	
- 1	Description

3

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets a value that indicates whether the validation control is enabled.

Use the **System.Web.UI.WebControls.BaseValidator.Enabled** property to specify whether the validation control is enabled. You can programmatically disable the validation control by setting this property to **false**.

EnableViewState

ErrorMessage

TrackViewState

## Description

Gets or sets the text for the error message.

Use the **System.Web.UI.WebControls.BaseValidator.ErrorMessage** property to specify the text to display in the validation control when validation fails. This text is also included in the

System.Web.UI.WebControls.ValidationSummary control, if one is placed on the Web page.

**Events** 

Font

ForeColor

**TrackViewState** 

Description

Gets or sets the color of the message displayed when validation fails.

Use the System.Web.UI.WebControls.BaseValidator.ForeColor
property to specify a custom text color for the message displayed when validation
fails.
HasChildViewState
Height
ID
IsTrackingViewState
IsValid
TrackViewState
Description
Gets or sets a value that indicates whether the associated input control
passes validation.
Use the System.Web.UI.WebControls.BaseValidator.IsValid property to
determine whether the associated input control passes validation.
NamingContainer
Page
Parent
PropertiesValid
TrackViewState
Description

2

3

4

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a value that indicates whether the control specified by the **System.Web.UI.WebControls.BaseValidator.ControlToValidate** property is a valid control.

Use the **System.Web.UI.WebControls.BaseValidator.PropertiesValid** property to determine whether the

System.Web.UI.WebControls.BaseValidator.ControlToValidate property contains a valid input control. To be a valid input control, the

**System.Web.UI.WebControls.BaseValidator.ControlToValidate** property must set to a control on the page.

RenderUplevel

**TrackViewState** 

[C#] protected bool RenderUplevel {get;}

[C++] protected: \_\_property bool get\_RenderUplevel();

[VB] Protected ReadOnly Property RenderUplevel As Boolean

[JScript] protected function get RenderUplevel(): Boolean;

Description

Gets a value that indicates whether the client's browser supports uplevel rendering.

Use the **System.Web.UI.WebControls.BaseValidator.RenderUplevel** property to determine whether the client's browser supports uplevel rendering. For a browser to support uplevel rendering, it must support Microsoft XML Document Object Model (DOM) version 4 or higher and any version of ECMAScript.

Site

1	Style
2	TabIndex
3	TagKey
4	TagName
5	TemplateSourceDirectory
6	Text
7	ToolTip
8	UniqueID
9	ViewState
10	ViewStateIgnoresCase
11	Visible
12	Width
13	AddAttributesToRender
14	
15	[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);
16	[C++] protected: void AddAttributesToRender(HtmlTextWriter* writer);
17	[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As
18	HtmlTextWriter)
	[IScript] protected override function AddAttributesToRender(writer:

Description

20

21

22

23

24

HtmlTextWriter);

Adds the attributes of this control to the output stream for rendering on the client. An **System.Web.UI.HtmlTextWriter** that contains the output stream to render on the client.

## CheckControlValidationProperty

[C#] protected void CheckControlValidationProperty(string name, string propertyName);

[C++] protected: void CheckControlValidationProperty(String\* name, String\* propertyName);

[VB] Protected Sub CheckControlValidationProperty(ByVal name As String, ByVal propertyName As String)

[JScript] protected function CheckControlValidationProperty(name : String, propertyName : String);

## Description

Helper function that verifies whether the specified control is on the page and contains validation properties.

The

System.Web.UI.WebControls.BaseValidator.CheckControlValidationPropert y(System.String,System.String) method is a helper function primarily used by the System.Web.UI.WebControls.BaseValidator.ControlPropertiesValid method to verify whether the specified control is on the page and contains validation properties. This method does not return a value to report the result. Instead, it throws an exception when verification fails. The control to verify. The property that contains the control name.

ControlPropertiesValid

[C#] protected virtual bool ControlPropertiesValid();

1	[C++] protected: virtual bool ControlPropertiesValid();
2	[VB] Overridable Protected Function ControlPropertiesValid() As Boolean
3	[JScript] protected function ControlPropertiesValid(): Boolean;
4	
5	Description
6	Helper function that determines whether the control specified by the
7	System.Web.UI.WebControls.BaseValidator.ControlToValidate property is a
8	valid control.
9	Return Value: true if the control specified by the
10	System.Web.UI.WebControls.BaseValidator.ControlToValidate property is a
11	valid control; otherwise, false.
12	The
13	System.Web.UI.WebControls.BaseValidator.ControlPropertiesValid method
14	is a helper function primarily used by the
15	System.Web.UI.WebControls.BaseValidator.PropertiesValid property to
16	determine whether the
17	System.Web.UI.WebControls.BaseValidator.ControlToValidate property
18	contains a valid input control. To be a valid input control, the
19	System.Web.UI.WebControls.BaseValidator.ControlToValidate property must
20	be set to some value and that value must be a control on the page.
21	DetermineRenderUplevel
22	
23	[C#] protected virtual bool DetermineRenderUplevel();
24	[C++] protected: virtual bool DetermineRenderUplevel();
25	[VB] Overridable Protected Function DetermineRenderUplevel() As Boolean

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[JScript] protected function DetermineRenderUplevel(): Boolean; Description Helper function that determines whether the validator control can be rendered for an uplevel browser. Return Value: true if the validator control can be rendered for an uplevel browser; otherwise, false. The System. Web. UI. WebControls. BaseValidator. Determine Render Uplevelmethod is a helper function used primarily by the  $System. Web. UI. Web Controls. Base Validator. Render Uplevel\ property\ to$ determine whether the client's browser supports uplevel rendering. For a browser to support uplevel rendering, it must support Microsoft XML Document Object Model (DOM) version 4 or higher and any version of ECMAScript. EvaluateIsValid [C#] protected abstract bool EvaluateIsValid(); [C++] protected: virtual bool EvaluateIsValid() = 0; [VB] MustOverride Protected Function EvaluateIsValid() As Boolean [JScript] protected abstract function EvaluateIsValid(): Boolean; Description

When overridden in a derived class, this method contains the code to determine whether the value in the input control is valid.

Return Value: true if the value in the input control is valid; otherwise, false.

2

3

4

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use the System.Web.UI.WebControls.BaseValidator.EvaluateIsValid method to determine whether the value in the input control specified by the System.Web.UI.WebControls.BaseValidator.ControlToValidate property is valid.

GetControlRenderID

[C#] protected string GetControlRenderID(string name);

[C++] protected: String\* GetControlRenderID(String\* name);

[VB] Protected Function GetControlRenderID(ByVal name As String) As String

[JScript] protected function GetControlRenderID(name : String) : String;

Description

Gets the client ID of the specified control.

Return Value: The client ID of the specified control.

Use the

System.Web.UI.WebControls.BaseValidator.GetControlRenderID(System.St ring) method to get the client ID of the specified control. The name of the control to get the client ID.

GetControlValidationValue

[C#] protected string GetControlValidationValue(string name);

[C++] protected: String\* GetControlValidationValue(String\* name);

[VB] Protected Function GetControlValidationValue(ByVal name As String) As

String

 $[JScript]\ protected\ function\ GetControlValidationValue (name:String):String;$ 

Description

Gets the value associated with the specified input control.

Return Value: The value associated with the specified input control.

Use the

System.Web.UI.WebControls.BaseValidator.GetControlValidationValue(System.String) method to get the value associated with the specified input control, regardless of the control type. For example, you can use this method to get the value in a System.Web.UI.WebControls.TextBox control, as well as the value of the selected item from a System.Web.UI.WebControls.ListBox control. The name of the input control to get the value from.

GetValidationProperty

[C#] public static PropertyDescriptor GetValidationProperty(object component); [C++] public: static PropertyDescriptor\* GetValidationProperty(Object\* component);

[VB] Public Shared Function GetValidationProperty(ByVal component As Object) As PropertyDescriptor

[JScript] public static function GetValidationProperty(component : Object) : PropertyDescriptor;

Description

Helper function to get the validation property of a control (if it exists).

Return Value: A System.ComponentModel.PropertyDescriptor that contains the validation property of the control.

1	The
2	System. Web. UI. Web Controls. Base Validator. Get Validation Property (System Controls) and the property of
3	Object) method is a helper function that gets the validation property of the
4	specified input control. An System.Object that represents the control to get the
5	validation property of.
6	OnInit
7	
8	[C#] protected override void OnInit(EventArgs e);
9	[C++] protected: void OnInit(EventArgs* e);
10	[VB] Overrides Protected Sub OnInit(ByVal e As EventArgs)
11	[JScript] protected override function OnInit(e : EventArgs);
12	
13	Description
14	Registers the validator on the page. An System.EventArgs that contains
15	the event data.
16	OnPreRender
17	
18	[C#] protected override void OnPreRender(EventArgs e);
19	[C++] protected: void OnPreRender(EventArgs* e);
20	[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)
21	[JScript] protected override function OnPreRender(e : EventArgs);
22	
23	Description
24	Checks the client brower and configures the validator for compatibility
25	prior to rendering.

1	OnUnload
2	
3	[C#] protected override void OnUnload(EventArgs e);
4	[C++] protected: void OnUnload(EventArgs* e);
5	[VB] Overrides Protected Sub OnUnload(ByVal e As EventArgs)
6	[JScript] protected override function OnUnload(e : EventArgs);
7	
8	Description
9	Un-registers the validator on the page.
10	RegisterValidatorCommonScript
11	
12	[C#] protected void RegisterValidatorCommonScript();
13	[C++] protected: void RegisterValidatorCommonScript();
14	[VB] Protected Sub RegisterValidatorCommonScript()
15	[JScript] protected function RegisterValidatorCommonScript();
16	
17	Description
18	Registers code on the page for client-side validation.
19	Use the
20	System.Web.UI.WebControls.BaseValidator.RegisterValidatorCommonScrip
21	t method to register code on the page to perform client-side validation.
22	RegisterValidatorDeclaration
23	
24	[C#] protected virtual void RegisterValidatorDeclaration();
25	[C++] protected: virtual void RegisterValidatorDeclaration();

1	[VB] Overridable Protected Sub RegisterValidatorDeclaration()
2	[JScript] protected function RegisterValidatorDeclaration(); Registers an
3	ECMAScript array declaration used on the page by client-side code.
4	
5	Description
6	Registers an ECMAScript array declaration using the array name,
7	Page_Validators .
8	Use the
9	System.Web.UI.WebControls.BaseValidator.RegisterValidatorDeclaration
10	method to register an ECMAScript array declaration using the array name,
11	Page_Validators .
12	Render
13	
14	[C#] protected override void Render(HtmlTextWriter writer);
15	[C++] protected: void Render(HtmlTextWriter* writer);
16	[VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)
17	[JScript] protected override function Render(writer : HtmlTextWriter);
18	
19	Description
20	Displays the control on the client. An System.Web.UI.HtmlTextWriter
21	that contains the output stream for rendering on the client.
22	Validate
23	
24	[C#] public void Validate();
25	[C++] public:sealed void Validate();

[VB] NotOverridable Public Sub Validate() [JScript] public function Validate(); 2 3 Description 4 Performs validation on the associated input control and updates the 5 System.Web.UI.WebControls.BaseValidator.IsValid property. 6 Use the System. Web. UI. Web Controls. Base Validator. Validate method to 7 perform validation on the associated input control. This method allows you to 8 programmatically perform validation on the input control. The 9  ${\bf System. Web. UI. WebControls. Base Validator. Is Valid}\ property\ is\ automatically$ 10 updated with the validation results. 11 BorderStyle enumeration (System.Web.UI.WebControls) 12 Validate 13 14 15 Description 16 Specifies the border style of a control. 17 The System.Web.UI.WebControls.BorderStyle enumeration represents 18 the different border style options for a control. 19 Validate 20 21 [C#] public const BorderStyle Dashed; 22 [C++] public: const BorderStyle Dashed; 23 [VB] Public Const Dashed As BorderStyle 24 [JScript] public var Dashed : BorderStyle;

	4	Validate
	5	
	6	[C#] public const BorderStyle Dotted;
	7	[C++] public: const BorderStyle Dotted;
	8	[VB] Public Const Dotted As BorderStyle
	9	[JScript] public var Dotted : BorderStyle;
	10	
	11	Description
	12	A dotted line border.
	13	Validate
	14	
	15	[C#] public const BorderStyle Double;
	16	[C++] public: const BorderStyle Double;
: mlig	17	[VB] Public Const Double As BorderStyle
	18	[JScript] public var Double : BorderStyle;
	19	
	20	Description
	21	A double solid line border.
	22	Validate

[C#] public const BorderStyle Groove;

[C++] public: const BorderStyle Groove;

A dashed line border.

Description

	1	[VB] Public Const Groove As BorderStyle
	2	[JScript] public var Groove : BorderStyle;
	3	
	4	Description
	5	A grooved border for a sunken border appearance.
	6	Validate
	7	
	8	[C#] public const BorderStyle Inset;
	9	[C++] public: const BorderStyle Inset;
	10	[VB] Public Const Inset As BorderStyle
	11	[JScript] public var Inset : BorderStyle;
	12	
	13	Description
į	14	An inset border for a sunken control appearance.
	15	Validate
	16	
•	17	[C#] public const BorderStyle None;
	18	[C++] public: const BorderStyle None;
	19	[VB] Public Const None As BorderStyle
	20	[JScript] public var None : BorderStyle;
	21	
	22	Description
	23	No border.
	24	Validate
	25	

1	
2	[C#] public const BorderStyle NotSet;
3	[C++] public: const BorderStyle NotSet;
4	[VB] Public Const NotSet As BorderStyle
5	[JScript] public var NotSet : BorderStyle;
6	
7	Description
8	No set border style.
9	Validate
10	
11	[C#] public const BorderStyle Outset;
12	[C++] public: const BorderStyle Outset;
13	[VB] Public Const Outset As BorderStyle
14	[JScript] public var Outset : BorderStyle;
15	
16	Description
17	An outset border for a raised control appearance.
18	Validate
19	
20	[C#] public const BorderStyle Ridge;
21	[C++] public: const BorderStyle Ridge;
22	[VB] Public Const Ridge As BorderStyle
23	[JScript] public var Ridge : BorderStyle;
24	
25	Description

23

24

25

A ridged border for a raised border appearance.

Validate

[C#] public const BorderStyle Solid;

[C++] public: const BorderStyle Solid;

[VB] Public Const Solid As BorderStyle

[JScript] public var Solid : BorderStyle;

Description

A solid line border.

BoundColumn class (System.Web.UI.WebControls)

**ToString** 

Description

A column type for the **System.Web.UI.WebControls.DataGrid** control that is bound to a field in a data source.

Use the **System.Web.UI.WebControls.BoundColumn** column type in a **System.Web.UI.WebControls.DataGrid** control to display the contents of a field in the data source in a single column. The field is linked to the

System.Web.UI.WebControls.BoundColumn , so any updates in the data source are reflected in the corresponding cells of the

 $System. Web. UI. Web Controls. Data Grid\ control.$ 

**ToString** 

	П	
	1	
	2	[C#] public static readonly string thisExpr;
	3	[C++] public: static String* thisExpr;
	4	[VB] Public Shared ReadOnly thisExpr As String
	5	[JScript] public static var thisExpr : String;
	6	
	7	Description
	8	Represents the string "!". This field is read-only.
	9	Use the System.Web.UI.WebControls.BoundColumn.thisExpr field to
201 <sub>1</sub>	10	represent the "!" string.
	11	BoundColumn
	12	Example Syntax:
	13	ToString
	14	
Sec. 18 (5 18 30 30 at al. 18 31 31 31 31 31 31 31 31 31 31 31 31 31	15	[C#] public BoundColumn();
74 74 74	16	[C++] public: BoundColumn();
e.Î de	17	[VB] Public Sub New()
	18	[JScript] public function BoundColumn();
	19	
	20	Description
	21	Initializes a new instance of the
	22	System.Web.UI.WebControls.BoundColumn class.
	23	Use this constructor to create and initialize a new instance of the
	24	System.Web.UI.WebControls.BoundColumn class.
	25	DataField
		II .

1	ToString
2	
3	[C#] public virtual string DataField {get; set;}
4	[C++] public:property virtual String* get_DataField();public:property
5	virtual void set_DataField(String*);
6	[VB] Overridable Public Property DataField As String
7	[JScript] public function get DataField(): String; public function set
8	DataField(String);
9	
10	Description
11	Gets or sets the field name from the data source to bind to the
12	System.Web.UI.WebControls.BoundColumn .
13	Use the System.Web.UI.WebControls.BoundColumn.DataField
14	property to specify the field to bind to the
15	System.Web.UI.WebControls.BoundColumn .
16	DataFormatString
17	ToString
18	
19	[C#] public virtual string DataFormatString {get; set;}
20	[C++] public:property virtual String* get_DataFormatString();public:
21	property virtual void set_DataFormatString(String*);
22	[VB] Overridable Public Property DataFormatString As String
23	[JScript] public function get DataFormatString(): String; public function set
24	DataFormatString(String);
25	

19

20

21

22

23

24

Description

2

3

5

Gets or sets the string that specifies the display format for items in the column.

Use the

System.Web.UI.WebControls.BoundColumn.DataFormatString property to provide a custom format for the items in the column.

DesignMode

FooterStyle

**FooterText** 

HeaderImageUrl

HeaderStyle

HeaderText

IsTrackingViewState

ItemStyle

Owner

ReadOnly

**ToString** 

Description

Gets or sets a value that indicates whether the items in the

System.Web.UI.WebControls.BoundColumn can be edited.

Use the **System.Web.UI.WebControls.BoundColumn.ReadOnly** property to specify whether the items in the

1	System. Web. UI. Web Controls. Bound Column can be edited. This property can
2	also be used to programmatically determine whether the column is read-only.
3	SortExpression
4	ViewState
5	Visible
6	FormatDataValue
7	
8	[C#] protected virtual string FormatDataValue(object dataValue);
9	[C++] protected: virtual String* FormatDataValue(Object* dataValue);
10	[VB] Overridable Protected Function FormatDataValue(ByVal dataValue As
11	Object) As String
12	[JScript] protected function FormatDataValue(dataValue : Object) : String;
13	
14	Description
15	Converts the specified value to the format indicated by the
16	System.Web.UI.WebControls.BoundColumn.DataFormatString property.
17	Return Value: The specified value converted to the format indicated by the
18	System.Web.UI.WebControls.BoundColumn.DataFormatString property.
19	Use the
20	System.Web.UI.WebControls.BoundColumn.FormatDataValue(System.Obje
21	ct) method to convert the specified value to the format indicated by the
22	System.Web.UI.WebControls.BoundColumn.DataFormatString property. The
23	value to format.
24	Initialize
25	

1	
1	
2	[C#] public override void Initialize();
3	[C++] public: void Initialize();
4	[VB] Overrides Public Sub Initialize()
5	[JScript] public override function Initialize();
6	
7	Description
8	Resets the System.Web.UI.WebControls.BoundColumn to its initial
9	state.
10	Use the System.Web.UI.WebControls.BoundColumn.Initialize method
11	to reset the System.Web.UI.WebControls.BoundColumn to its initial state.
12	InitializeCell
13	
14	[C#] public override void InitializeCell(TableCell cell, int columnIndex,
15	ListItemType itemType);
16	[C++] public: void InitializeCell(TableCell* cell, int columnIndex, ListItemType
17	itemType);
18	[VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal
19	columnIndex As Integer, ByVal itemType As ListItemType)
20	[JScript] public override function InitializeCell(cell: TableCell, columnIndex: int
21	itemType : ListItemType);
22	
23	Description
24	Resets the specified cell in the
25	System.Web.UI.WebControls.BoundColumn to its initial state.

5
6
7
8
9
10
11
12
13
14
15

The
System. Web. UI. Web Controls. Bound Column. Initialize Cell (System. Web. UI. Web
eb Controls. Table Cell, System. Int 32, System. Web. UI. Web Controls. List I tem Tyrum Cells and Cells
pe) method resets the specified cell in the
System.Web.UI.WebControls.BoundColumn to its initial state. A
System.Web.UI.WebControls.TableCell object that represents the cell to reset.
The column number where the cell is located. One of the
System.Web.UI.WebControls.ListItemType values.
Button class (System.Web.UI.WebControls)
TrackViewState
Description
Displays a push button control on the Web page.
Use the System.Web.UI.WebControls.Button control to create a push
button on the Web page. You can create either a submit button or a command
button.
Button
Example Syntax:
TrackViewState
[C#] public Button();
[C++] public: Button();
[VB] Public Sub New()
[JScript] public function Button();

$\Gamma$			, •	
De	500	71 N	111	าท
$\mathcal{L}_{\mathcal{L}}$	$\sigma c_i$	$\nu \nu$	**	,,,

5

7

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the **System.Web.UI.WebControls.Button** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.Button** class.

AccessKey

Attributes

BackColor

BorderColor

BorderStyle

**BorderWidth** 

CausesValidation

**TrackViewState** 

## Description

Gets or sets a value indicating whether validation is performed when the **System.Web.UI.WebControls.Button** control is clicked.

By default, page validation is performed when a

page all pass the validation rules specified by the validation control.

**System.Web.UI.WebControls.Button** control is clicked. Page validation determines whether the input controls associated with a validation control on the

**ChildControlsCreated** 

ClientID

CommandArgument
TrackViewState

Description

2

5

6

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets an optional parameter passed to the **System.Web.UI.WebControls.Button.Command** event along with the associated **System.Web.UI.WebControls.Button.CommandName**.

Use the **System.Web.UI.WebControls.Button.CommandArgument** property to specify an parameter that compliments the

System.Web.UI.WebControls.Button.CommandName property.

CommandName

**TrackViewState** 

[C#] public string CommandName {get; set;}

[C++] public: \_\_property String\* get\_CommandName();public: \_\_property void set CommandName(String\*);

[VB] Public Property CommandName As String

[JScript] public function get CommandName(): String;public function set

CommandName(String);

Description

Gets or sets the command name associated with the

System.Web.UI.WebControls.Button control that is passed to the

System.Web.UI.WebControls.Button.Command event.

24

25

1

2

3

5 6 7 8 Context 9 Controls 10 ControlStyle 11 12 CssClass 13 Enabled 14 EnableViewState 15 **Events** 16 **Font** 17 ForeColor 18 **HasChildViewState** 19 Height 20 ID 21 IsTrackingViewState 22

NamingContainer

Page

Parent

When you have multiple System. Web. UI. Web Controls. Button controls on a Web page, use the System. Web.UI. WebControls. Button. CommandName property to specify or determine the command name associated with each System.Web.UI.WebControls.Button control. You can set the System.Web.UI.WebControls.Button.CommandName property with any string that identifies the command to perform. You can then programmatically determine the command name of the System. Web.UI. WebControls. Button control and perform the appropriate actions. ControlStyleCreated

13

14

15

16

17

18

19

20

21

22

23

Style 2 TabIndex 3 TagKey TagName TemplateSourceDirectory Text **TrackViewState** 9 10 11

Site

Description

Gets or sets the text caption displayed in the

System.Web.UI.WebControls.Button control.

Use the System.Web.UI.WebControls.Button.Text property to specify or determine the caption to display in the System.Web.UI.WebControls.Button control.

ToolTip

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

TrackViewState

24 25 Description

5

10

11

13

14

15

16

17

18

19

20

21

23

24

Occurs when the **System.Web.UI.WebControls.Button** control is clicked.

The **System.Web.UI.WebControls.Button.Click** event is raised when the **System.Web.UI.WebControls.Button** control is clicked. This event is commonly used when no command name is associated with the

**System.Web.UI.WebControls.Button** control (for instance, with a **submit** button).

**TrackViewState** 

[C#] public event CommandEventHandler Command;

[C++] public: event CommandEventHandler\* Command;

[VB] Public Event Command As CommandEventHandler

Description

Occurs when the System. Web. UI. WebControls. Button control is clicked.

The System.Web.UI.WebControls.Button control is clicked. This event is commonly used when a command name, such as Sort, is associated with the System.Web.UI.WebControls.Button control. This allows you to create multiple System.Web.UI.WebControls.Button controls on a Web page and programmatically determine which System.Web.UI.WebControls.Button control is clicked.

AddAttributesToRender

1	
2	[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);
3	[C++] protected: void AddAttributesToRender(HtmlTextWriter* writer);
4	[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As
5	HtmlTextWriter)
6	[JScript] protected override function AddAttributesToRender(writer:
7	HtmlTextWriter);
8	
9	Description
10	Adds the attributes of the System.Web.UI.WebControls.Button control to
11	the output stream for rendering on the client. An
12	System.Web.UI.HtmlTextWriter that contains the output stream to render on the
13	client.
14	OnClick
15	
16	[C#] protected virtual void OnClick(EventArgs e);
17	[C++] protected: virtual void OnClick(EventArgs* e);
18	[VB] Overridable Protected Sub OnClick(ByVal e As EventArgs)
19	[JScript] protected function OnClick(e: EventArgs);
20	
21	Description
22	Raises the System.Web.UI.WebControls.Button.Click event of the
23	System.Web.UI.WebControls.Button control.
24	The System.Web.UI.WebControls.Button.Click event is raised when the
25	System.Web.UI.WebControls.Button control is clicked. This event is commonly

3

4

5

7

8

9

10

11

12

13

14

15

17

19

20

21

22

23

24

used when no command name is associated with the

System.Web.UI.WebControls.Button control (for instance, with a submit

button). A System. Event Args that contains the event data.

**OnCommand** 

[C#] protected virtual void OnCommand(CommandEventArgs e);

[C++] protected: virtual void OnCommand(CommandEventArgs\* e);

[VB] Overridable Protected Sub OnCommand(ByVal e As CommandEventArgs)

[JScript] protected function OnCommand(e: CommandEventArgs);

Description

Raises the **System.Web.UI.WebControls.Button.Command** event of the **System.Web.UI.WebControls.Button** control.

The System.Web.UI.WebControls.Button.Command event is raised when the System.Web.UI.WebControls.Button control is clicked. This event is commonly used when a command name, such as Sort, is associated with the System.Web.UI.WebControls.Button control. This allows you to create multiple System.Web.UI.WebControls.Button controls on a Web page and programmatically determine which System.Web.UI.WebControls.Button control is clicked. A System.Web.UI.WebControls.CommandEventArgs that contains the event data.

RenderContents

[C#] protected override void RenderContents(HtmlTextWriter writer);

[C++] protected: void RenderContents(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

[JScript] protected override function RenderContents(writer : HtmlTextWriter);

Description

IPostBackEventHandler.RaisePostBackEvent

[C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument); [C++] void IPostBackEventHandler::RaisePostBackEvent(String\* eventArgument);

[VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements

IPostBackEventHandler.RaisePostBackEvent

IISorint | function | IPostBackEventHandler | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEvent(eventArgument) | PaigePostBackEv

[JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument : String);

ButtonColumn class (System.Web.UI.WebControls)

**TrackViewState** 

#### Description

A column type for the **System.Web.UI.WebControls.DataGrid** control that contains a user-defined command button, such as **Add** or **Remove**, that corresponds with each row in the column.

Use the System.Web.UI.WebControls.ButtonColumn column type in a System.Web.UI.WebControls.DataGrid control to create a command button that corresponds with each row in the System.Web.UI.WebControls.DataGrid

```
control. Specify the caption displayed in the command buttons by setting the
    System.Web.UI.WebControls.ButtonColumn.Text property.
           ButtonColumn
3
          Example Syntax:
           TrackViewState
5
6
    [C#] public ButtonColumn();
    [C++] public: ButtonColumn();
    [VB] Public Sub New()
    [JScript] public function ButtonColumn();
10
11
    Description
12
           Initializes a new instance of the
13
    System.Web.UI.WebControls.ButtonColumn class.
14
           Use this constructor to create and initialize a new instance of the
15
    System.Web.UI.WebControls.ButtonColumn class.
16
           ButtonType
17
           TrackViewState
18
19
    [C#] public virtual ButtonColumnType ButtonType {get; set;}
20
    [C++] public: __property virtual ButtonColumnType get ButtonType();public:
21
      _property virtual void set_ButtonType(ButtonColumnType);
22
    [VB] Overridable Public Property ButtonType As ButtonColumnType
23
    [JScript] public function get ButtonType(): ButtonColumnType; public function
24
    set ButtonType(ButtonColumnType);
25
```

### Description

1

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets the type of button to display in the System.Web.UI.WebControls.ButtonColumn column.

Use this property to specify whether the buttons in the column are displayed as link or push buttons. Clicking on the command buttons in the column raises the **System.Web.UI.WebControls.DataGrid.ItemCommand** event. You can programmatically control the action performed when the command button is clicked by providing an event handler for the

System.Web.UI.WebControls.DataGrid.ItemCommand event.

CommandName

**TrackViewState** 

[C#] public virtual string CommandName {get; set;}

[C++] public: \_\_property virtual String\* get\_CommandName();public: \_\_property virtual void set\_CommandName(String\*);

[VB] Overridable Public Property CommandName As String

[JScript] public function get CommandName(): String; public function set

CommandName(String);

#### Description

Gets or sets a string that represents the command to perform when a button in the System.Web.UI.WebControls.ButtonColumn is clicked.

Use the System.Web.UI.WebControls.ButtonColumn.CommandName property to associate a command name, such as Add or Remove, with a button.

You can set the System.Web.UI.WebControls.ButtonColumn.CommandName property to any string that identifies the action to perform when the command button is clicked. You can then programmatically determine the command name in the event handler for the System.Web.UI.WebControls.DataGrid.ItemCommand event and perform the appropriate actions.

DataTextField

TrackViewState

[C#] public virtual string DataTextField {get; set;}

[C++] public: \_\_property virtual String\* get\_DataTextField();public: \_\_property virtual void set DataTextField(String\*);

[VB] Overridable Public Property DataTextField As String

[JScript] public function get DataTextField(): String;public function set DataTextField(String);

Description

Gets or sets the field name from a data source to bind to the System.Web.UI.WebControls.ButtonColumn .

Use the System.Web.UI.WebControls.ButtonColumn.DataTextField property to specify the field name from the data source to bind to the System.Web.UI.WebControls.ButtonColumn.Text property of the buttons in the System.Web.UI.WebControls.ButtonColumn.

DataTextFormatString

**TrackViewState** 

21

22

23

24

25

1 [C#] public virtual string DataTextFormatString {get; set;} [C++] public: property virtual String\* get DataTextFormatString();public: 3 property virtual void set DataTextFormatString(String\*); 4 [VB] Overridable Public Property DataTextFormatString As String 5 [JScript] public function get DataTextFormatString(): String; public function set DataTextFormatString(String); 8 Description 9 Gets or sets the string that specifies the display format for the caption in 10 each command button. 11 Use the 12 System.Web.UI.WebControls.ButtonColumn.DataTextFormatString property 13 to provide a custom display format for the caption of the command buttons in the column. 15 DesignMode 16 FooterStyle 17 FooterText 18 HeaderImageUrl 19

FooterText
HeaderImageUrl
HeaderStyle
HeaderText
IsTrackingViewState
ItemStyle

SortExpression

Owner

Text **TrackViewState** 

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Description

Gets or sets the caption displayed in the command buttons of the System.Web.UI.WebControls.ButtonColumn .

Use the System. Web. UI. Web Controls. Button Column. Text property to specify or determine the caption displayed in the command buttons of the  $System. Web. UI. Web Controls. Button Column \ .$ 

ViewState

Visible

FormatDataTextValue

[C#] protected virtual string FormatDataTextValue(object dataTextValue);

[C++] protected: virtual String\* FormatDataTextValue(Object\* dataTextValue);

[VB] Overridable Protected Function FormatDataTextValue(ByVal

dataTextValue As Object) As String

[JScript] protected function FormatDataTextValue(dataTextValue : Object) :

String;

Description

Converts the specified value to the format indicated by the

System.Web.UI.WebControls.ButtonColumn.DataTextFormatString property.

Return Value: The specified value converted to the format indicated by the System.Web.UI.WebControls.ButtonColumn.DataTextFormatString 2 property. 3 Use the System.Web.UI.WebControls.ButtonColumn.FormatDataTextValue(System. 5 **Object)** method to convert the specified value to the format indicated by the 6 System.Web.UI.WebControls.ButtonColumn.DataTextFormatString 7 property. The value to format. Initialize 9 10 [C#] public override void Initialize(); 11 [C++] public: void Initialize(); 12 [VB] Overrides Public Sub Initialize() 13 [JScript] public override function Initialize(); 14 15 Description 16 Resets the System. Web. UI. Web Controls. Button Column to its initial 17 state. 18 Use the System. Web. UI. Web Controls. Button Column. Initialize method 19 to reset the System. Web.UI. WebControls. Button Column to its initial state. 20 **InitializeCell** 21 22 [C#] public override void InitializeCell(TableCell cell, int columnIndex, 23 ListItemType itemType); [C++] public: void InitializeCell(TableCell\* cell, int columnIndex, ListItemType

1	itemType);
2	[VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal
3	columnIndex As Integer, ByVal itemType As ListItemType)
4	[JScript] public override function InitializeCell(cell: TableCell, columnIndex: int,
5	itemType : ListItemType);
6	
7	Description
8	Initializes a cell in the System.Web.UI.WebControls.ButtonColumn to
9	its initial state.
10	The
11	System.Web.UI.WebControls.ButtonColumn.InitializeCell(System.Web.UI.
12	WebControls.TableCell,System.Int32,System.Web.UI.WebControls.ListItem
13	Type) method resets the specified cell in the
14	System.Web.UI.WebControls.ButtonColumn to its initial state. A
15	System.Web.UI.WebControls.TableCell object that represents the cell to reset.
16	The column number where the cell is located. One of the
17	System.Web.UI.WebControls.ListItemType values.
18	ButtonColumnType enumeration (System.Web.UI.WebControls)
19	TrackViewState
20	
21	
22	Description
23	Specifies the button type for the
24	System.Web.UI.WebControls.ButtonColumn object.
25	

2

24

25

The System.Web.UI.WebControls.ButtonColumnType enumeration represents the button styles for the System.Web.UI.WebControls.ButtonColumn object. [C#] public const ButtonColumnType LinkButton; [C++] public: const ButtonColumnType LinkButton; [VB] Public Const LinkButton As ButtonColumnType [JScript] public var LinkButton: ButtonColumnType; A column of hyperlink style buttons. [C#] public const ButtonColumnType PushButton; [C++] public: const ButtonColumnType PushButton; [VB] Public Const PushButton As ButtonColumnType [JScript] public var PushButton : ButtonColumnType; Calendar class (System.Web.UI.WebControls) **ToString** 

# Description

3

6

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Displays a single month calendar that allows the user to select dates and move to the next or previous month.

Use the **System.Web.UI.WebControls.Calendar** control to display a single month of a calendar on a Web page. The control allows you to select dates and move to the next or previous month. The

System.Web.UI.WebControls.Calendar control supports all of the

System.Globalization.Calendar types in the System.Globalization namespace.

Apart from the Gregorian calendar, this also includes calendars that use different year and month systems, such as the Hjiri calendar.

Calendar

Example Syntax:

**ToString** 

[C#] public Calendar();

[C++] public: Calendar();

[VB] Public Sub New()

[JScript] public function Calendar();

Description

Initializes a new instance of the **System.Web.UI.WebControls.Calendar** class.

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.Calendar class. 2 AccessKey 3 Attributes BackColor BorderColor BorderStyle BorderWidth CellPadding 9 **ToString** 10 11 12 Description 13 Gets or sets the amount of space between the contents of a cell and the 14 cell's border. 15 Use this property to control the spacing between the contents of a cell and 16 the cell's border. The padding amount specified is added to all four sides of a cell. 17 Individual cell sizes cannot be specified. 18 CellSpacing 19 **ToString** 20 21 [C#] public int CellSpacing {get; set;} 22 [C++] public: \_\_property int get CellSpacing();public: \_\_property void 23 set CellSpacing(int); 24 [VB] Public Property CellSpacing As Integer

ToString

1	[JScript] public function get CellSpacing(): int;public function set
2	CellSpacing(int);
3	
4	Description
5	Gets or sets the amount of space between cells.
6	Use this property to control the spacing between individual cells in the
7	calendar. This spacing is applied both vertically and horizontally.
8	ChildControlsCreated
9	ClientID
10	Context
11	Controls
12	ControlStyle
13	ControlStyleCreated
14	CssClass
15	DayHeaderStyle
16	ToString
17	
18	
19	Description
20	Gets the style properties for the section that displays the day of the week.
21	Use this property to specify the style for the section that displays the days
22	of the week. For additional information on the different style properties that can be
23	controlled, see System.Web.UI.WebControls.TableItemStyle.
24	DayNameFormat

A Table 2 1971; Study Study Trees 1985

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

[C#] public DayNameFormat DayNameFormat {get; set;}
[C++] public: \_\_property DayNameFormat get\_DayNameFormat();public:
\_\_property void set\_DayNameFormat(DayNameFormat);
[VB] Public Property DayNameFormat As DayNameFormat
[JScript] public function get DayNameFormat() : DayNameFormat;public function

Description

Gets or sets the name format of days of the week.

set DayNameFormat(DayNameFormat);

Use the **System.Web.UI.WebControls.Calendar.DayNameFormat** property to specify the name format for the days of the week. This property is set with one of the **System.Web.UI.WebControls.DayNameFormat** enumeration values. You can specify whether the days of the week are displayed as the full name, short (abbreviated) name, first letter of the day, or first two letters of the day.

DayStyle

**ToString** 

[C#] public TableItemStyle DayStyle {get;}

[C++] public: property TableItemStyle\* get DayStyle();

[VB] Public ReadOnly Property DayStyle As TableItemStyle

[JScript] public function get DayStyle(): TableItemStyle;

Description

20

21

22

23

24

25

Gets the style properties for the days in the displayed month. Use this property to specify the style for the days in the displayed month. Enabled 3 **EnableViewState Events** FirstDayOfWeek **ToString** 9 Description 10 Gets or sets the day of the week to display in the first day column of the 11 System.Web.UI.WebControls.Calendar control. 12 Use the System. Web. UI. Web Controls. Calendar. First Day Of Week 13 property to specify the day of the week to display in the first day column of the 14 System.Web.UI.WebControls.Calendar control. This property is set with one of 15 the System. Web. UI. Web Controls. First Day Of Week enumeration values. You 16 can specify any day of the week or FirstDayOfWeek.Default, which indicates 17 that the date is determined by the system settings. 18

Font

ForeColor

HasChildViewState

Height

ID

IsTrackingViewState

NamingContainer

NextMonthText

**ToString** 

Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the text displayed for the next month navigation control.

Use the **System.Web.UI.WebControls.Calendar.NextMonthText** property to provide custom text for the next month navigation control. This property is commonly used along with the

**System.Web.UI.WebControls.Calendar.PrevMonthText** property to provide a custom set of navigation controls.

NextPrevFormat

**ToString** 

[C#] public NextPrevFormat NextPrevFormat {get; set;}

[C++] public: \_\_property NextPrevFormat get\_NextPrevFormat();public:

property void set NextPrevFormat(NextPrevFormat);

[VB] Public Property NextPrevFormat As NextPrevFormat

[JScript] public function get NextPrevFormat(): NextPrevFormat; public function set NextPrevFormat(NextPrevFormat);

Description

Gets or sets the format of the next and previous month navigation elements in the title section of the **System.Web.UI.WebControls.Calendar** control.

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use the System. Web. UI. Web Controls. Calendar. Next Prev Format 1 property to specify the format for the next and previous month navigation elements on the System. Web. UI. WebControls. Calendar control. This property is set with one of the System.Web.UI.WebControls.NextPrevFormat enumeration values. You can specify whether the text for the navigation controls is displayed as the full month name, the short (abbreviated) month name, or custom text. NextPrevStyle **ToString** [C#] public TableItemStyle NextPrevStyle {get;} [C++] public: property TableItemStyle\* get NextPrevStyle(); [VB] Public ReadOnly Property NextPrevStyle As TableItemStyle

[JScript] public function get NextPrevStyle(): TableItemStyle;

Description

Gets the style properties for the next and previous month navigation elements.

Use this property to specify the style for the next and previous month navigation elements. For additional information on the different style properties that can be controlled, see System. Web. UI. Web Controls. Table I tem Style.

OtherMonthDayStyle

**ToString** 

[C#] public TableItemStyle OtherMonthDayStyle {get;}

22

23

24

25

[C++] public: \_\_property TableItemStyle\* get\_OtherMonthDayStyle();[VB] Public ReadOnly Property OtherMonthDayStyle As TableItemStyle[JScript] public function get OtherMonthDayStyle(): TableItemStyle;

Description

Gets the style properties for the days on the System.Web.UI.WebControls.Calendar control that are not in the displayed month.

Use this property to specify the style for the days on the System.Web.UI.WebControls.Calendar control that are not in the displayed month. If the System.Web.UI.WebControls.Calendar.OtherMonthDayStyle property is not set, the style specified in the

System.Web.UI.WebControls.Calendar.DayStyle property is used.

Page

Parent

PrevMonthText

**ToString** 

Description

Gets or sets the text displayed for the previous month navigation control.

Use the **System.Web.UI.WebControls.Calendar.PrevMonthText**property to provide custom text for the previous month navigation control. This
property is commonly used along with the

1	System.Web.UI.WebControls.Calendar.NextMonthText property to provide a
2	custom set of navigation controls.
3	SelectedDate
4	ToString
5	
6	[C#] public DateTime SelectedDate {get; set;}
7	[C++] public:property DateTime get_SelectedDate();public:property void
8	set_SelectedDate(DateTime);
9	[VB] Public Property SelectedDate As DateTime
10	[JScript] public function get SelectedDate(): DateTime; public function set
11	SelectedDate(DateTime);
12	
13	Description
14	Gets or sets the selected date.
15	Use the System.Web.UI.WebControls.Calendar.SelectedDate property
16	to determine the selected date on the System.Web.UI.WebControls.Calendar
17	control. The System.Web.UI.WebControls.Calendar.SelectedDate property is
18	typically used when the System.Web.UI.WebControls.Calendar.SelectionMode
19	property is set to CalendarSelectionMode.Day. This setting only allows a single
20	date selection from the System.Web.UI.WebControls.Calendar control.
21	SelectedDates
22	ToString
23	
24	[C#] public SelectedDatesCollection SelectedDates {get;}
25	[C++] public:property SelectedDatesCollection* get_SelectedDates();

25

[VB] Public ReadOnly Property SelectedDates As SelectedDatesCollection [JScript] public function get SelectedDates(): SelectedDatesCollection; 3 Description Gets a collection of System. Date Time objects that represent the selected 5 dates on the System. Web. UI. Web Controls. Calendar control. Use the System. Web. UI. Web Controls. Calendar. Selected Dates 7 collection to determine the currently selected dates on the System.Web.UI.WebControls.Calendar control. This property is typically used when the System. Web. UI. WebControls. Calendar. Selection Mode property is set 10 to Calendar Selection Mode. Day Week or 11 CalendarSelectionMode.DayWeekMonth . These settings allow you to select 12 multiple dates from the System. Web. UI. Web Controls. Calendar control by week 13 or month. 14 SelectedDayStyle 15 **ToString** 16 17 [C#] public TableItemStyle SelectedDayStyle {get;} 18 [C++] public: property TableItemStyle\* get\_SelectedDayStyle(); 19 [VB] Public ReadOnly Property SelectedDayStyle As TableItemStyle 20 [JScript] public function get SelectedDayStyle(): TableItemStyle; 21 22 Description 23 Gets the style properties for the selected dates.

1629 MS1-863US.APP lee@hayes plic 509-324-9256

Use this property to specify the style for the selected dates on the System.Web.UI.WebControls.Calendar control. If the System.Web.UI.WebControls.Calendar.SelectedDayStyle property is not set, the style specified in the System.Web.UI.WebControls.Calendar.DayStyle property is used.

SelectionMode

**ToString** 

[C#] public CalendarSelectionMode SelectionMode {get; set;}
[C++] public: \_\_property CalendarSelectionMode get\_SelectionMode();public:
\_\_property void set\_SelectionMode(CalendarSelectionMode);
[VB] Public Property SelectionMode As CalendarSelectionMode
[JScript] public function get SelectionMode() : CalendarSelectionMode;public function set SelectionMode(CalendarSelectionMode);

Description

Gets or sets the date selection mode on the

**System.Web.UI.WebControls.Calendar** control that specifies whether the user can select a single day, a week, or an entire month.

Use the **System.Web.UI.WebControls.Calendar.SelectionMode** property to specify the date selection mode on the

**System.Web.UI.WebControls.Calendar** control. This property is set with one of the **System.Web.UI.WebControls.CalendarSelectionMode** enumeration values.

You can specify whether the user can select a single day, a week, or an entire month, or you can disable date selection capability entirely.

1	SelectMonthText
2	ToString
3	
4	[C#] public string SelectMonthText {get; set;}
5	[C++] public:property String* get_SelectMonthText();public:property void
6	set_SelectMonthText(String*);
7	[VB] Public Property SelectMonthText As String
8	[JScript] public function get SelectMonthText(): String; public function set
9	SelectMonthText(String);
10	
11	Description
12	Gets or sets the text displayed for the month selection element in the
13	selector column.
14	Use the System.Web.UI.WebControls.Calendar.SelectMonthText
15	property to provide custom text for the month selection element in the selector
16	column.
17	SelectorStyle
18	ToString
19	
20	[C#] public TableItemStyle SelectorStyle {get;}
21	[C++] public:property TableItemStyle* get_SelectorStyle();
22	[VB] Public ReadOnly Property SelectorStyle As TableItemStyle
23	[JScript] public function get SelectorStyle() : TableItemStyle;
24	
25	Description

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

2

Gets the style properties for the week and month selector column.

Use this property to specify the style for the week and month selector column. For additional information on the different style properties that can be controlled, see System.Web.UI.WebControls.TableItemStyle.

SelectWeekText

**ToString** 

[C#] public string SelectWeekText {get; set;}

[C++] public: \_\_property String\* get\_SelectWeekText();public: \_\_property void set SelectWeekText(String\*);

[VB] Public Property SelectWeekText As String

[JScript] public function get SelectWeekText() : String; public function set SelectWeekText(String);

Description

Gets or sets the text displayed for the week selection element in the selector column.

Use the **System.Web.UI.WebControls.Calendar.SelectWeekText** property to provide custom text for the week selection element in the selector column.

ShowDayHeader

**ToString** 

[C#] public bool ShowDayHeader {get; set;}

[C++] public: \_\_property bool get\_ShowDayHeader();public: \_\_property void

Į,	and Chara Davids and davids and the
1	set_ShowDayHeader(bool);
2	[VB] Public Property ShowDayHeader As Boolean
3	[JScript] public function get ShowDayHeader(): Boolean; public function set
4	ShowDayHeader(Boolean);
5	
6	Description
7	Gets or sets a value indicating whether the heading for the days of the week
8	is displayed.
9	Use the System.Web.UI.WebControls.Calendar.ShowDayHeader
10	property to specify whether the heading for the days of the week is displayed.
11	ShowGridLines
12	ToString
13	
14	[C#] public bool ShowGridLines {get; set;}
15	[C++] public:property bool get_ShowGridLines();public:property void
16	set_ShowGridLines(bool);
17	[VB] Public Property ShowGridLines As Boolean
18	[JScript] public function get ShowGridLines(): Boolean; public function set
19	ShowGridLines(Boolean);
20	
21	Description
22	Gets or sets a value indicating whether the days on the
23	System.Web.UI.WebControls.Calendar control are separated with grid lines.
24	
25	

1	Use the System.Web.UI.WebControls.Calendar.ShowGridLines
2	property to specify whether the days on the
3	System.Web.UI.WebControls.Calendar control are separated with grid lines.
4	ShowNextPrevMonth
5	ToString
6	
7	[C#] public bool ShowNextPrevMonth {get; set;}
8	[C++] public:property bool get_ShowNextPrevMonth();public:property
9	<pre>void set_ShowNextPrevMonth(bool);</pre>
10	[VB] Public Property ShowNextPrevMonth As Boolean
11	[JScript] public function get ShowNextPrevMonth(): Boolean; public function set
12	ShowNextPrevMonth(Boolean);
13	
14	Description
15	Gets or sets a value indicating whether the
16	System.Web.UI.WebControls.Calendar control displays the next and previous
17	month navigation elements in the title section.
18	Use the System.Web.UI.WebControls.Calendar.ShowNextPrevMonth
19	property to specify whether the next and previous month navigation elements are
20	displayed in the title section.
21	ShowTitle
22	ToString
23	
24	[C#] public bool ShowTitle {get; set;}
25	[C++] public:property bool get_ShowTitle();public:property void

25

set ShowTitle(bool); [VB] Public Property ShowTitle As Boolean [JScript] public function get ShowTitle(): Boolean; public function set 3 ShowTitle(Boolean); Description 6 Gets or sets a value indicating whether the title section is displayed. Use the System. Web.UI. WebControls. Calendar. Show Title property to 8 specify whether the title section is displayed. 9 Site 10 Style 11 TabIndex 12 TagKey 13 **TagName** 14 **TemplateSourceDirectory** 15 **TitleFormat** 16 **ToString** 17 18 19 Description 20 Gets or sets the title format for the title section. 21 Use the System. Web.UI. WebControls. Calendar. Title Format property to 22 specify the format for the title section. This property is set with one of the 23 System.Web.UI.WebControls.TitleFormat enumeration values. You can specify

whether the title displays as the month or both the month and the year.

TitleStyle **ToString** 2 [C#] public TableItemStyle TitleStyle {get;} [C++] public: property TableItemStyle\* get TitleStyle(); [VB] Public ReadOnly Property TitleStyle As TableItemStyle [JScript] public function get TitleStyle(): TableItemStyle; 7 8 Description 9 Gets the style properties of the title heading for the 10 System.Web.UI.WebControls.Calendar control. 11 Use this property to specify the style for the title heading of the 12 System.Web.UI.WebControls.Calendar control. For additional information on 13 the different style properties that can be controlled, see 14 System.Web.UI.WebControls.TableItemStyle . 15 TodayDayStyle 16 **ToString** 17 18 [C#] public TableItemStyle TodayDayStyle {get;} 19 [C++] public: property TableItemStyle\* get TodayDayStyle(); 20 [VB] Public ReadOnly Property TodayDayStyle As TableItemStyle 21 [JScript] public function get TodayDayStyle(): TableItemStyle; 22 23 Description 24 25

1	Gets the style properties for today's date on the
2	System.Web.UI.WebControls.Calendar control.
3	Use this property to specify the style for today's date on the
4	System.Web.UI.WebControls.Calendar control. If the
5	System.Web.UI.WebControls.Calendar.TodayDayStyle property is not set, the
6	style specified in the System.Web.UI.WebControls.Calendar.DayStyle property
7	is used.
8	TodaysDate
9	ToString
10	
11	[C#] public DateTime TodaysDate {get; set;}
12	[C++] public:property DateTime get_TodaysDate();public:property void
13	set_TodaysDate(DateTime);
14	[VB] Public Property TodaysDate As DateTime
15	[JScript] public function get TodaysDate(): DateTime;public function set
16	TodaysDate(DateTime);
17	
18	Description
19	Gets or sets the value for today's date.
20	Use the System.Web.UI.WebControls.Calendar.TodaysDate property to
21	determine today's date. You can also use this property to programmatically set the
22	value for today's date on the System.Web.UI.WebControls.Calendar control.
23	This property is set using a <b>System.DateTime</b> object.
24	ToolTip
25	UniqueID

1	ViewState
2	ViewStateIgnoresCase
3	Visible
4	VisibleDate
5	ToString
6	
7	
8	Description
9	Gets or sets the date that specifies the month to display on the
10	System.Web.UI.WebControls.Calendar control.
11	Use the System.Web.UI.WebControls.Calendar.VisibleDate property to
12	programmatically set the month to display on the
13	System.Web.UI.WebControls.Calendar control. The specified date can be any
14	date in the month you want to display, although it is common to use the first date
15	in the month. This property is set using a <b>System.DateTime</b> object.
16	WeekendDayStyle
17	ToString
18	
19	[C#] public TableItemStyle WeekendDayStyle {get;}
20	[C++] public:property TableItemStyle* get_WeekendDayStyle();
21	[VB] Public ReadOnly Property WeekendDayStyle As TableItemStyle
22	[JScript] public function get WeekendDayStyle() : TableItemStyle;
23	
24	Description

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets the style properties for the weekend dates on the **System.Web.UI.WebControls.Calendar** control.

System.Web.UI.WebControls.Calendar.ontrol. If the System.Web.UI.WebControls.Calendar.WeekendDayStyle property is not set, the style specified in the System.Web.UI.WebControls.Calendar.DayStyle property is used.

Use this property to specify the style for the weekend dates on the

Width

**ToString** 

### Description

Occurs when each day is created in the control hierarchy for the **System.Web.UI.WebControls.Calendar** control.

This event is raised when each day is created in the control hierarchy for the System.Web.UI.WebControls.Calendar control.

**ToString** 

### Description

Occurs when the user selects a day, a week, or an entire month by clicking the date selector controls.

This event is raised when the user selects a day, a week, or an entire month by clicking the date selector controls.

**ToString** 

Descriptio	n

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Occurs when the user clicks on the next or previous month navigation controls on the title heading.

This event is raised when the user clicks on the next or previous month navigation elements on the title heading.

CreateControlCollection

[C#] protected override ControlCollection CreateControlCollection();

[C++] protected: ControlCollection\* CreateControlCollection();

[VB] Overrides Protected Function CreateControlCollection() As

ControlCollection

[JScript] protected override function CreateControlCollection():

ControlCollection;

Description

HasWeekSelectors

[C#] protected bool HasWeekSelectors(CalendarSelectionMode selectionMode);

[C++] protected: bool HasWeekSelectors(CalendarSelectionMode

selectionMode);

[VB] Protected Function HasWeekSelectors(ByVal selectionMode As

CalendarSelectionMode) As Boolean

1	[JScript] protected function HasWeekSelectors(selectionMode:
2	CalendarSelectionMode): Boolean;
3	
4	Description
5	Determines if a System.Web.UI.WebControls.CalendarSelectionMode
6	contains week selectors.
7	Return Value: true if the
8	System.Web.UI.WebControls.CalendarSelectionMode contains week selectors
9	otherwise false.
10	System.Web.UI.WebControls.CalendarSelectionMode values of
11	DayWeek and DayWeekMonth contains week selectors. One of the
12	System.Web.UI.WebControls.CalendarSelectionMode values.
13	LoadViewState
14	
15	[C#] protected override void LoadViewState(object savedState);
16	[C++] protected: void LoadViewState(Object* savedState);
17	[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)
18	[JScript] protected override function LoadViewState(savedState : Object);
19	
20	Description
21	Loads a saved state of the System.Web.UI.WebControls.Calendar . An
22	System.Object that contains the saved condition of the
23	System.Web.UI.WebControls.Calendar.
24	OnDayRender
25	

[C#] protected virtual void OnDayRender(TableCell cell, CalendarDay day);
[C++] protected: virtual void OnDayRender(TableCell\* cell, CalendarDay\* day);
[VB] Overridable Protected Sub OnDayRender(ByVal cell As TableCell, ByVal day As CalendarDay)

[JScript] protected function OnDayRender(cell: TableCell, day: CalendarDay);

### Description

Raises the System.Web.UI.WebControls.Calendar.DayRender event of the System.Web.UI.WebControls.Calendar control and allows you to provide a custom handler for the System.Web.UI.WebControls.Calendar.DayRender event.

Although databinding is not supported for the

System.Web.UI.WebControls.Calendar control, it is possible modify the

content and formatting of the individual date cells. Before the

System.Web.UI.WebControls.Calendar control is displayed on the Web page, it

creates and assembles the components that make up the control. The

System.Web.UI.WebControls.Calendar.DayRender event is raised when each
date cell in System.Web.UI.WebControls.Calendar control is created. You can
control the contents and formatting of a date cell when it is created by providing
code in the event handler for the

System.Web.UI.WebControls.Calendar.DayRender event. A

System.Web.UI.WebControls.TableCell that contains information about the cell to render. A System.Web.UI.WebControls.CalendarDay that contains information about the day to render.

OnSel	ection	Chan	oed
OHOO	(CCHOII	Chan;	<b>XUU</b>

[C#] protected virtual void OnSelectionChanged();

[C++] protected: virtual void OnSelectionChanged();

[VB] Overridable Protected Sub OnSelectionChanged()

[JScript] protected function OnSelectionChanged();

### Description

2

7

8

9

10

11

12

13

14

15

16

17

18

20

21

22

23

24

Raises the System.Web.UI.WebControls.Calendar.SelectionChanged event of the System.Web.UI.WebControls.Calendar control and allows you to provide a custom handler for the

System.Web.UI.WebControls.Calendar.SelectionChanged event.

The **System.Web.UI.WebControls.Calendar.SelectionChanged** event is raised when the user selects a day, a week, or an entire month by clicking the date selector controls.

**OnVisibleMonthChanged** 

[C#] protected virtual void OnVisibleMonthChanged(DateTime newDate,

DateTime previousDate);

[C++] protected: virtual void OnVisibleMonthChanged(DateTime newDate,

DateTime previousDate);

[VB] Overridable Protected Sub OnVisibleMonthChanged(ByVal newDate As

DateTime, ByVal previousDate As DateTime)

[JScript] protected function OnVisibleMonthChanged(newDate: DateTime,

previousDate : DateTime);

System
System
System
System
System
System

3

10 11

9

13

14

12

15

16

17 18

19

20

21 22

23 24

25

Description

Raises the

System.Web.UI.WebControls.Calendar.VisibleMonthChanged event of the System.Web.UI.WebControls.Calendar control and allows you to provide a custom handler for the

System.Web.UI.WebControls.Calendar.VisibleMonthChanged event.

The System.Web.UI.WebControls.Calendar.VisibleMonthChanged event is raised when the user clicks on the next or previous month navigation elements on the title section. A System.DateTime object that represents the month currently displayed in the System.Web.UI.WebControls.Calendar control. A System.DateTime object that represents the previous month displayed by the System.Web.UI.WebControls.Calendar control.

Render

[C#] protected override void Render(HtmlTextWriter writer);

[C++] protected: void Render(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)

[JScript] protected override function Render(writer: HtmlTextWriter);

Description

Displays the **System.Web.UI.WebControls.Calendar** control on the client. An **System.Web.UI.HtmlTextWriter** that contains the output stream for rendering on the client.

SaveViewState

1	
2	[C#] protected override object SaveViewState();
3	[C++] protected: Object* SaveViewState();
4	[VB] Overrides Protected Function SaveViewState() As Object
5	[JScript] protected override function SaveViewState() : Object;
6	
7	Description
8	Stores the state of the System.Web.UI.WebControls.Calendar.
9	Return Value: An object that contains the saved state of the
10	System.Web.UI.WebControls.Calendar .
11	IPostBackEventHandler.RaisePostBackEvent
12	
13	[C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);
14	[C++] void IPostBackEventHandler::RaisePostBackEvent(String*
15	eventArgument);
16	[VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements
17	IPostBackEventHandler.RaisePostBackEvent
18	[JScript] function IPsstBackEventHandler.RaisePostBackEvent(eventArgument:
19	String);
20	TrackViewState
21	
22	[C#] protected override void TrackViewState();
23	[C++] protected: void TrackViewState();
24	[VB] Overrides Protected Sub TrackViewState()
25	[JScript] protected override function TrackViewState():

G

Description

Mark

3

5

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Marks the starting point to begin tracking and saving changes to the control as part of the control viewstate.

CalendarDay class (System.Web.UI.WebControls)

**TrackViewState** 

## Description

Represents a date in the System.Web.UI.WebControls.Calendar control.

A System.Web.UI.WebControls.Calendar control. You can use this class in the System.Web.UI.WebControls.Calendar.DayRender event handler to programmatically access the properties of a date as it is rendered on the System.Web.UI.WebControls.Calendar control. This allows you to determine the properties of the day (such as whether the date is selectable, selected, today's date, or a weekend date) and programmatically control the appearance or behavior of the day.

CalendarDay

Example Syntax:

TrackViewState

[C#] public CalendarDay(DateTime date, bool isWeekend, bool isToday, bool isSelected, bool isOtherMonth, string dayNumberText);

[C++] public: CalendarDay(DateTime date, bool isWeekend, bool isToday, bool

isSelected, bool isOtherMonth, String\* dayNumberText);

[VB] Public Sub New(ByVal date As DateTime, ByVal isWeekend As Boolean, ByVal isToday As Boolean, ByVal isSelected As Boolean, ByVal isOtherMonth As Boolean, ByVal dayNumberText As String)

[JScript] public function CalendarDay(date: DateTime, isWeekend: Boolean, isToday: Boolean, isSelected: Boolean, isOtherMonth: Boolean, dayNumberText: String);

### Description

Initializes a new instance of the

System. Web. UI. WebControls. Calendar Day class.

The following table shows initial property values for an instance of System.Web.UI.WebControls.CalendarDay . A System.DateTime object that contains the date represented by an instance of this class. true to indicate that the date represented by an instance of this class is either a Saturday or a Sunday; otherwise, false. true to indicate that the date represented by an instance of this class is the current date; otherwise, false. true to indicate that the date represented by an instance of this class is selected on the

System.Web.UI.WebControls.Calendar control; otherwise, false. true to indicate that the date represented by an instance of this class is in a month other than the displayed month on the System.Web.UI.WebControls.Calendar control; otherwise, false. The day number for the date represented by this class.

Date

TrackViewState

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] public DateTime Date {get;} [C++] public: property DateTime get Date(); 3 [VB] Public ReadOnly Property Date As DateTime [JScript] public function get Date(): DateTime; 5 6 Description 7 8 9

Gets the date represented by an instance of this class. This property is readonly.

Use the System. Web. UI. Web Controls. Calendar Day. Date property to programmatically determine the date represented by an instance of this class.

DayNumberText

**TrackViewState** 

[C#] public string DayNumberText {get;}

[C++] public: \_\_property String\* get DayNumberText();

[VB] Public ReadOnly Property DayNumberText As String

[JScript] public function get DayNumberText(): String;

# Description

Gets the string equivalent of the day number for the date represented by an instance of the System. Web. UI. WebControls. Calendar Day class. This property is read-only.

Use the System. Web. UI. Web Controls. Calendar Day. Day Number Text property to determine the string equivalent of the day number for the date

represented by an instance of this class. This allows you to programmatically 1 control the appearance or behavior of the day, based on this value. 2 **IsOtherMonth** 3 **TrackViewState** 5 [C#] public bool IsOtherMonth {get;} 6 [C++] public: property bool get IsOtherMonth(); 7 [VB] Public ReadOnly Property IsOtherMonth As Boolean 8 [JScript] public function get IsOtherMonth(): Boolean; 9 10 Description 11 Gets a value that indicates whether the date represented by an instance of 12 this class is in a month other than the month displayed in the 13 System.Web.UI.WebControls.Calendar control. This property is read-only. 14 Use the System. Web. UI. WebControls. Calendar Day. Is Other Month 15 property to programmatically determine whether the date represented by an 16 instance of this class is in a month other than the month displayed in the 17 System.Web.UI.WebControls.Calendar control. This allows you to 18 programmatically control the appearance or behavior of the day based on this 19 value. 20 **IsSelectable** 21 **TrackViewState** 22 23

[C#] public bool IsSelectable {get; set;}

[C++] public: \_\_property bool get\_IsSelectable();public: \_\_property void

set\_IsSelectable(bool); 1 [VB] Public Property IsSelectable As Boolean 2 [JScript] public function get IsSelectable(): Boolean; public function set 3 IsSelectable(Boolean); 5 Description 6 Gets or sets a value that indicates whether the date represented by an 7 instance of this class can be selected in the 8 System. Web. UI. Web Controls. Calendar control. 9 Use the System. Web. UI. Web Controls. Calendar Day. Is Selectable 10 property to specify or determine whether the date represented by an instance of 11 this class can be selected in the System. Web.UI. WebControls. Calendar control. 12 This allows you to programmatically control the appearance behavior of the day, 13 based on this value. 14 **IsSelected** 15 **TrackViewState** 16 17 [C#] public bool IsSelected {get;} 18 [C++] public: \_\_property bool get IsSelected(); 19 [VB] Public ReadOnly Property IsSelected As Boolean 20 [JScript] public function get IsSelected(): Boolean; 21 22 Description 23 24 25

1 |

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a value that indicates whether the date represented by an instance of this class is selected in the **System.Web.UI.WebControls.Calendar** control. This property is read-only.

Use the **System.Web.UI.WebControls.CalendarDay.IsSelected** property to programmatically determine whether the date represented by an instance of this class is selected in the **System.Web.UI.WebControls.Calendar** control. This allows you to programmatically control the appearance or behavior of the day, based on this value.

IsToday

**TrackViewState** 

[C#] public bool IsToday {get;}

[C++] public: \_\_property bool get\_IsToday();

[VB] Public ReadOnly Property IsToday As Boolean

[JScript] public function get IsToday(): Boolean;

## Description

Gets a value that indicates whether the date represented by an instance of this class is the same date specified by the

System.Web.UI.WebControls.Calendar.TodaysDate property of the System.Web.UI.WebControls.Calendar control. This property is read-only.

Use the **System.Web.UI.WebControls.CalendarDay.IsToday** property to programmatically determine whether the date represented by an instance of this class is the same date specified by the

 ${\bf System. Web. UI. WebControls. Calendar. To days Date}\ property\ of\ the$ 

System.Web.UI.WebControls.Calendar control. This allows you to programmatically control the appearance or behavior of the day, based on this value.

IsWeekend

TrackViewState

[C#] public bool IsWeekend {get;}

[C++] public: \_property bool get\_IsWeekend();

[VB] Public ReadOnly Property IsWeekend As Boolean

[JScript] public function get IsWeekend(): Boolean;

Description

Gets a value that indicates whether the date represented by an instance of this class is a either Saturday or Sunday. This property is read-only.

Use the **System.Web.UI.WebControls.CalendarDay.IsWeekend**property to programmatically determine whether the date represented by an
instance of this class is either a Saturday or a Sunday. This allows you to
programmatically control the appearance or behavior of the day, based on this
value.

CalendarSelectionMode enumeration (System.Web.UI.WebControls)
ToString

Description

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Specifies the date selection mode of the System.Web.UI.WebControls.Calendar control.  $The \ System. Web. UI. Web Controls. Calendar Selection Mode \ enumeration$ represents the date selection modes of the System.Web.UI.WebControls.Calendar control. **ToString** [C#] public const CalendarSelectionMode Day; [C++] public: const CalendarSelectionMode Day; [VB] Public Const Day As Calendar Selection Mode [JScript] public var Day: CalendarSelectionMode; Description A single date can be selected on the System.Web.UI.WebControls.Calendar control. **ToString** [C#] public const CalendarSelectionMode DayWeek; [C++] public: const CalendarSelectionMode DayWeek; [VB] Public Const DayWeek As CalendarSelectionMode [JScript] public var DayWeek: CalendarSelectionMode; Description

A single day or entire week can be selected on the System.Web.UI.WebControls.Calendar control.

1	ToString
2	
3	[C#] public const CalendarSelectionMode DayWeekMonth;
4	[C++] public: const CalendarSelectionMode DayWeekMonth;
5	[VB] Public Const DayWeekMonth As CalendarSelectionMode
6	[JScript] public var DayWeekMonth : CalendarSelectionMode;
7	
8	Description
9	A single date, week, or entire month can be selected on the
10	System.Web.UI.WebControls.Calendar control.
11	ToString
12	
13	[C#] public const CalendarSelectionMode None;
14	[C++] public: const CalendarSelectionMode None;
15	[VB] Public Const None As CalendarSelectionMode
16	[JScript] public var None: CalendarSelectionMode;
17	
18	Description
19	No dates can be selected on the System.Web.UI.WebControls.Calendar
20	control.
21	TableRow.CellControlCollection class (System.Web.UI.WebControls)
22	ToString
23	
24	
25	Description

Count 2 IsReadOnly IsSynchronized Item Owner SyncRoot Add 9 [C#] public override void Add(Control child); 10 [C++] public: void Add(Control\* child); 11 [VB] Overrides Public Sub Add(ByVal child As Control) 12 [JScript] public override function Add(child : Control); 13 14 Description 15 Adds the specified System. Web. UI. Control object to the collection. The 16 new control is added to the end of the array. 17 AddAt 18 19 [C#] public override void AddAt(int index, Control child); 20 [C++] public: void AddAt(int index, Control\* child); 21 [VB] Overrides Public Sub AddAt(ByVal index As Integer, ByVal child As 22 Control) 23 [JScript] public override function AddAt(index : int, child : Control); 24 25

Description

2

3

4

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Adds the specified **System.Web.UI.Control** object to the collection. The new control is added to the array at the specified index location. The location in the array to add the child control. The **Control** object to add to the collection.

CheckBox class (System.Web.UI.WebControls)

**ToString** 

Description

Displays a check box that allows the user to select a true or false condition.

Use the **System.Web.UI.WebControls.CheckBox** control to allow the user to select a **true** or **false** state.

CheckBox

Example Syntax:

**ToString** 

[C#] public CheckBox();

[C++] public: CheckBox();

[VB] Public Sub New()

[JScript] public function CheckBox();

Description

Initializes a new instance of the **System.Web.UI.WebControls.CheckBox** class.

25

Use this constructor to create and initialize a new instance of the  $System. Web. UI. Web Controls. Check Box \ class.$ 2 AccessKey 3 Attributes AutoPostBack **ToString** 8 Description 9 Gets or sets a value indicating whether the 10  ${\bf System. Web. UI. WebControls. CheckBox} \ {\bf state} \ {\bf automatically} \ posts \ {\bf back} \ to \ the$ 11 server when clicked. 12 Use this property to specify whether the state of the 13 System.Web.UI.WebControls.CheckBox control is posted back to the server 14 when clicked. 15 BackColor 16 BorderColor 17 BorderStyle 18 BorderWidth 19 Checked 20 **ToString** 21 22 23 Description

	1	Gets or sets a value indicating whether the
	2	System.Web.UI.WebControls.CheckBox control is checked.
	3	Use this property to determine whether the
	4	System.Web.UI.WebControls.CheckBox control is checked. This property can
	5	also be used to programmatically set the state of the
	6	System.Web.UI.WebControls.CheckBox control.
	7	ChildControlsCreated
	8	ClientID
	9	Context
i i	10	Controls
	11	ControlStyle.
	12	ControlStyleCreated
Sood tank that from hold had that bad	13	CssClass
The state of the s	14	Enabled
	15	EnableViewState
	16	Events
:	17	Font
	18	ForeColor
	19	HasChildViewState
	20	Height
	21	ID
	22	IsTrackingViewState
	23	NamingContainer
	24	Page
	25	Parent

Site Style 2 **TabIndex** 3 TagKey TagName 5 **TemplateSourceDirectory** Text **ToString** 8 9 10 Description 11 Gets or sets the text label associated with the 12 System.Web.UI.WebControls.CheckBox . 13 Use this property to specify the text label associated with the 14 System.Web.UI.WebControls.CheckBox control. This property can also be used 15 to programmatically get the text label associated with the 16 System.Web.UI.WebControls.CheckBox control. 17 **TextAlign** 18 **ToString** 19 20 [C#] public virtual TextAlign TextAlign {get; set;} 21 [C++] public: property virtual TextAlign get TextAlign();public: \_\_property 22 virtual void set TextAlign(TextAlign); 23 [VB] Overridable Public Property TextAlign As TextAlign 24 [JScript] public function get TextAlign(): TextAlign; public function set

TextAlign(TextAlign);

Description

2

3

4

5

6

7

8

9

10

11

12

19

21

24

25

Gets or sets the alignment of the text label associated with the System.Web.UI.WebControls.CheckBox control.

Use this property to specify the alignment of the text label associated with the System. Web. UI. Web Controls. Check Box control. You can specify whether the text label appears to the right or left of the check box. Use the System.Web.UI.WebControls.CheckBox.Text property to specify the label text.

ToolTip

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

**ToString** 

Description

 ${\tt Occurs\ when\ the\ System. Web. UI. WebControls. CheckBox. Checked}$ property is changed.

The System.Web.UI.WebControls.CheckBox.CheckedChanged event is raised when the System. Web. UI. WebControls. CheckBox. Checked property is changed.

OnCheckedChanged

1 [C#] protected virtual void OnCheckedChanged(EventArgs e); [C++] protected: virtual void OnCheckedChanged(EventArgs\* e); 3 [VB] Overridable Protected Sub OnCheckedChanged(ByVal e As EventArgs) [JScript] protected function OnCheckedChanged(e: EventArgs); 5 6 Description 7 Raises the System. Web. UI. Web Controls. Check Box. Checked Changed 8 event of the System. Web.UI. WebControls. CheckBox control. This allows you 9 to handle the event directly. 10 The System.Web.UI.WebControls.CheckBox.CheckedChanged event is 11 raised when the System. Web. UI. WebControls. CheckBox. Checked property is 12 changed. A System. EventArgs that contains the event data. 13 OnPreRender 14 15 [C#] protected override void OnPreRender(EventArgs e); 16 [C++] protected: void OnPreRender(EventArgs\* e); 17 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs) 18 [JScript] protected override function OnPreRender(e: EventArgs); 19 20 Description 21 Registers client script for generating postback prior to rendering on the 22  $client\ if\ System. Web. UI. WebControls. CheckBox. AutoPostBack\ is\ true\ .$ 23 Render 24

[C#] protected override void Render(HtmlTextWriter writer); [C++] protected: void Render(HtmlTextWriter\* writer); 3 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter) [JScript] protected override function Render(writer: HtmlTextWriter); 5 6 Description 7 Displays the System. Web. UI. Web Controls. Check Box on the client. A 8 System. Web. UI. Html Text Writer that contains the output stream to render on the 9 client. 10 IPostBackDataHandler.LoadPostData 11 12 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey, 13 NameValueCollection postCollection); 14 [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey, 15 NameValueCollection\* postCollection); 16 [VB] Function LoadPostData(ByVal postDataKey As String, ByVal 17 postCollection As NameValueCollection) As Boolean Implements 18 IPostBackDataHandler.LoadPostData 19 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey: String, 20 postCollection: NameValueCollection): Boolean; 21 IPostBackDataHandler.RaisePostDataChangedEvent 22 23 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent(); 24 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

[VB] Sub RaisePostDataChangedEvent() Implements

IPostBackDataHandler.RaisePostDataChangedEvent

[JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();

CheckBoxList class (System.Web.UI.WebControls)

TrackViewState

Description

Creates a multi selection check box group that can be dynamically created by binding the control to a data source.

The System.Web.UI.WebControls.CheckBoxList control provides a multi selection check box group that can be dynamically generated with data binding. It contains an System.Web.UI.WebControls.ListControl.Items collection with members that correspond to individual items in the list. To determine which items are checked, loop through and test the System.Web.UI.WebControls.ListItem.Selected property of each item in the list.

CheckBoxList

Example Syntax:

TrackViewState

[C#] public CheckBoxList();

[C++] public: CheckBoxList();

[VB] Public Sub New()

[JScript] public function CheckBoxList();

Description

1

2

3

4

5

6

7

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the

System.Web.UI.WebControls.CheckBoxList class.

Use this constructor to create and initialize a new instance of the

 $System. Web. UI. Web Controls. Check Box List \ class.$ 

AccessKey

Attributes

AutoPostBack

BackColor

BorderColor

BorderStyle

BorderWidth

CellPadding

TrackViewState

Description

Gets or sets the distance (in pixels) between the border and contents of the cell.

Use this property to control the spacing between the contents of a cell and the cell's border in the **System.Web.UI.WebControls.CheckBoxList** control.

CellSpacing

TrackViewState

1	
2	[C#] public virtual int CellSpacing {get; set;}
3	[C++] public:property virtual int get_CellSpacing();public:property virtual
4	<pre>void set_CellSpacing(int);</pre>
5	[VB] Overridable Public Property CellSpacing As Integer
6	[JScript] public function get CellSpacing(): int;public function set
7	CellSpacing(int);
8	
9	Description
10	Gets or sets the distance (in pixels) between cells.
11	Use this property to control the spacing between individual cells in the
12	System.Web.UI.WebControls.CheckBoxList control. This property is applied
13	both vertically and horizontally.
14	ChildControlsCreated
15	ClientID
16	Context
17	Controls
18	ControlStyle
19	ControlStyleCreated
20	CssClass
21	DataMember
22	DataSource
23	DataTextField
24	DataTextFormatString
25	DataValueField

2

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Enabled EnableViewState **Events** Font ForeColor HasChildViewState Height ID IsTrackingViewState Items NamingContainer Page

Parent

RepeatColumns

TrackViewState

### Description

Gets or sets the number of columns to display in the System.Web.UI.WebControls.CheckBoxList control.

Use this property to specify the number of columns that display items in the System.Web.UI.WebControls.CheckBoxList control. If this property is not set, the System.Web.UI.WebControls.CheckBoxList control displays all list items in a single column.

RepeatDirection

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Tracl	7-1	7:	A11	,C	to	40	,
Trac	кι	/ 1	СN	$\sim$	14	Ιt	7

[C#1	public	virtual R	epeatDirection	RepeatDirection	{get; set;}
------	--------	-----------	----------------	-----------------	-------------

[C++] public: \_\_property virtual RepeatDirection get\_RepeatDirection();public:

property virtual void set\_RepeatDirection(RepeatDirection);

[VB] Overridable Public Property RepeatDirection As RepeatDirection

[JScript] public function get RepeatDirection(): RepeatDirection; public function set RepeatDirection(RepeatDirection);

### Description

Gets or sets a value that indicates whether the control displays vertically or horizontally.

Use this property to specify the display direction of the **System.Web.UI.WebControls.CheckBoxList** control.

RepeatLayout

**TrackViewState** 

[C#] public virtual RepeatLayout RepeatLayout {get; set;}

[C++] public: \_\_property virtual RepeatLayout get\_RepeatLayout();public:

\_\_property virtual void set\_RepeatLayout(RepeatLayout);

[VB] Overridable Public Property RepeatLayout As RepeatLayout

[JScript] public function get RepeatLayout(): RepeatLayout; public function set

RepeatLayout(RepeatLayout);

Description

22

23

24

25

1

Gets or sets the layout of the check boxes.

Use this property to specify whether the items in the

System.Web.UI.WebControls.CheckBoxList control are displayed in a table. If this property is set to RepeatLayout.Table, the items in the list are displayed in a table. If this property is set to RepeatLayout.Flow, the items in the list are displayed without a table structure.

SelectedIndex

SelectedItem

Site

Style

**TabIndex** 

TagKey

TagName

**TemplateSourceDirectory** 

**TextAlign** 

TrackViewState

### Description

Gets or sets the text alignment for the check boxes within the group.

Use this property to specify whether the text associated with the check boxes appears on the left or right of the check box. If this property is set to **TextAlign.Right**, the text is displayed to the right of the check box. If this property is set to **TextAlign.Left**, the text is displayed to the left of the check box.

ToolTip

1	UniqueID
2	ViewState
3	ViewStateIgnoresCase
4	Visible
5	Width
6	CreateControlStyle
7	
8	[C#] protected override Style CreateControlStyle();
9	[C++] protected: Style* CreateControlStyle();
10	[VB] Overrides Protected Function CreateControlStyle() As Style
11	[JScript] protected override function CreateControlStyle(): Style;
12	
13	Description
14	Creates a new control style object.
15	Return Value: A System.Web.UI.WebControls.Style that contains the style
16	properties of a control.
17	FindControl
18	
19	[C#] protected override Control FindControl(string id, int pathOffset);
20	[C++] protected: Control* FindControl(String* id, int pathOffset);
21	[VB] Overrides Protected Function FindControl(ByVal id As String, ByVal
22	pathOffset As Integer) As Control
23	[JScript] protected override function FindControl(id : String, pathOffset : int) :
24	Control;
25	

1 Description 2 Catches post data for each System.Web.UI.WebControls.CheckBox in 3 the list. OnPreRender 5 6 [C#] protected override void OnPreRender(EventArgs e); 7 [C++] protected: void OnPreRender(EventArgs\* e); 8 [VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs) 9 [JScript] protected override function OnPreRender(e: EventArgs); 10 11 Description 12 Configures the System. Web. UI. Web Controls. Check Box List prior to 13 rendering on the client. 14 Render 15 16 [C#] protected override void Render(HtmlTextWriter writer); 17 [C++] protected: void Render(HtmlTextWriter\* writer); 18 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter) 19 [JScript] protected override function Render(writer: HtmlTextWriter); 20 21 Description 22 Displays the System. Web. UI. Web Controls. Check Box List on the client. 23 A System. Web.UI. HtmlTextWriter that contains the output stream for rendering 24 on the client. 25

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

IPoctRa	ckDat	aHar	dler I	_oadPostDa	ta
TEOSIDA	UKIJAL	ariar	IUICI . L	Juaur Osiba	ıu

[C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
NameValueCollection postCollection);
[C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,

NameValueCollection\* postCollection);

[VB] Function LoadPostData(ByVal postDataKey As String, ByVal postCollection As NameValueCollection) As Boolean Implements

IPostBackDataHandler.LoadPostData

 $[JScript]\ function\ IPostBackDataHandler. LoadPostData(postDataKey: String, and the property of the propert$ 

postCollection: NameValueCollection): Boolean;

IPostBackDataHandler. RaisePostDataChangedEvent

[C#] void IPostBackDataHandler.RaisePostDataChangedEvent();

[C++] void IPostBackDataHandler::RaisePostDataChangedEvent();

[VB] Sub RaisePostDataChangedEvent() Implements

IPostBackDataHandler. RaisePostDataChangedEvent

 $[JScript]\ function\ IPostBackDataHandler. RaisePostDataChangedEvent();$ 

IRepeatInfoUser.GetItemStyle

[C#] Style IRepeatInfoUser.GetItemStyle(ListItemType itemType, int repeatIndex);

[C++] Style\* IRepeatInfoUser::GetItemStyle(ListItemType itemType, int repeatIndex);

 $[VB]\ Function\ GetItemStyle (ByVal\ itemType\ As\ ListItemType,\ ByVal\ itemType)$ 

1	repeatIndex As Integer) As Style Implements IRepeatInfoUser.GetItemStyle
2	[JScript] function IRepeatInfoUser.GetItemStyle(itemType : ListItemType,
3	repeatIndex : int) : Style;
4	IRepeatInfoUser.RenderItem
5	
6	[C#] void IRepeatInfoUser.RenderItem(ListItemType itemType, int repeatIndex
7	RepeatInfo repeatInfo, HtmlTextWriter writer);
8	[C++] void IRepeatInfoUser::RenderItem(ListItemType itemType, int
9	repeatIndex, RepeatInfo* repeatInfo, HtmlTextWriter* writer);
10	[VB] Sub RenderItem(ByVal itemType As ListItemType, ByVal repeatIndex As
11	Integer, ByVal repeatInfo As RepeatInfo, ByVal writer As HtmlTextWriter)
12	Implements IRepeatInfoUser.RenderItem
13	[JScript] function IRepeatInfoUser.RenderItem(itemType : ListItemType,
14	repeatIndex: int, repeatInfo: RepeatInfo, writer: HtmlTextWriter);
15	CommandEventArgs class (System.Web.UI.WebControls)
16	TrackViewState
17	
18	
19	Description
20	Provides data for the <b>Command</b> event.
21	The <b>Command</b> event is raised when a
22	System.Web.UI.WebControls.Button or
23	System.Web.UI.WebControls.ImageButton control is clicked.
24	CommandEventArgs
25	Example Syntax:

7	Trac	kV	iew	Sta	te
1	1100	. N	14,700	1 7 1.61	

1

3

4

5

6

7

8

9

10

11

13 14

15 16

17

18

19

21

20

22

23

24

25

[C#] public CommandEventArgs(CommandEventArgs e);

[C++] public: CommandEventArgs(CommandEventArgs\* e);

[VB] Public Sub New(ByVal e As CommandEventArgs)

[JScript] public function CommandEventArgs(e: CommandEventArgs);

Initializes a new instance of the

 $System. Web. UI. Web Controls. Command Event Args\ class.$ 

Description

Initializes a new instance of the

System.Web.UI.WebControls.CommandEventArgs class with another System.Web.UI.WebControls.CommandEventArgs object.

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.CommandEventArgs class using the specified System.Web.UI.WebControls.CommandEventArgs object. A System.Web.UI.WebControls.CommandEventArgs that contains the event data.

CommandEventArgs

Example Syntax:

**TrackViewState** 

[C#] public CommandEventArgs(string commandName, object argument);

[C++] public: CommandEventArgs(String\* commandName, Object\* argument);

[VB] Public Sub New(ByVal commandName As String, ByVal argument As

1	Object)
2	[JScript] public function CommandEventArgs(commandName : String, argument :
3	Object);
4	
5	Description
6	Initializes a new instance of the
7	System.Web.UI.WebControls.CommandEventArgs class with the specified
8	command name and argument.
9	Use this constructor to create and initialize a new instance of the
10	System.Web.UI.WebControls.CommandEventArgs class using the specified
11	command name and argument. The name of the command. A System.Object that
12	contains the arguments for the command.
13	CommandArgument
14	TrackViewState
15	
16	[C#] public object CommandArgument {get;}
17	[C++] public:property Object* get_CommandArgument();
18	[VB] Public ReadOnly Property CommandArgument As Object
19	[JScript] public function get CommandArgument(): Object;
20	
21	Description
22	Gets the argument for the command.
23	The
24	System.Web.UI.WebControls.CommandEventArgs.CommandArgument can
25	contain any string set by the programmer. The

1	System. Web. UI. Web Controls. Command Event Args. Command Argument
2	property complements the
3	$System. Web. UI. Web Controls. Command Event Args. Command Name\ property$
4	by allowing you to provide any additional information for the command. For
5	example, you can set the
6	System.Web.UI.WebControls.CommandEventArgs.CommandName property
7	to Sort and set the
8	System.Web.UI.WebControls.CommandEventArgs.CommandArgument
9	property to Ascending to specify a command to sort in ascending order.
10	CommandName
11	TrackViewState
12	
13	[C#] public string CommandName {get;}
14	[C++] public:property String* get_CommandName();
15	[VB] Public ReadOnly Property CommandName As String
16	[JScript] public function get CommandName(): String;
17	
18	Description
19	Gets the name of the command.
20	Use the
21	System.Web.UI.WebControls.CommandEventArgs.CommandName property
22	to determine the command to perform. The
23	System.Web.UI.WebControls.CommandEventArgs.CommandName property
24	can contain any string set by the programmer. The programmer can then identify
25	the command name in code and perform the appropriate tasks.

CommandEventHandler delegate (System.Web.UI.WebControls)
ToString

Description

Represents the method that will handle the **Command** event. The source of the event. A **System.Web.UI.WebControls.CommandEventArgs** that contains the event data.

The Command event is raised when a

System.Web.UI.WebControls.Button or

System.Web.UI.WebControls.ImageButton control is clicked.

CompareValidator class (System.Web.UI.WebControls)

**ToString** 

Description

Compares the value entered by the user into an input control with the value entered into another input control or a constant value.

Use the System.Web.UI.WebControls.CompareValidator control to compare the value entered by the user into an input control, such as a System.Web.UI.WebControls.TextBox control, with the value entered into another input control or a constant value. You can also use the System.Web.UI.WebControls.CompareValidator control to indicate whether the value entered into an input control can be converted to the data type specified by the System.Web.UI.WebControls.BaseCompareValidator.Type property.

	1	CompareValidator
	2	Example Syntax:
	3	ToString
	4	
	5	[C#] public CompareValidator();
	6	[C++] public: CompareValidator();
	7	[VB] Public Sub New()
	8	[JScript] public function CompareValidator();
	9	AccessKey
<b>-</b>	10	Attributes
	11	BackColor
	12	BorderColor
	13	BorderStyle
	14	BorderWidth
the state that the state that	15	ChildControlsCreated
	16	ClientID
:: :::::::::::::::::::::::::::::::::::	17	Context
	18	Controls
	19	ControlStyle
	20	ControlStyleCreated
	21	ControlToCompare
	22	ToString
	23	
	24	
	25	Description

Gets or sets the input control to compare with the input control being 1 validated. 2 Use the 3 System. Web. UI. Web Controls. Compare Validator. Control To Compareproperty to specify an input control, such as a 5  ${\bf System. Web. UI. WebControls. TextBox}\ ,\ to\ compare\ with\ the\ input\ control\ being$ 6 validated. If the input control specified by this property is not a control on the page, an exception is thrown. 8 ControlToValidate 9 CssClass 10 Display 11 EnableClientScript 12 Enabled 13 **EnableViewState** 14 ErrorMessage 15 **Events** 16 Font 17 ForeColor 18 HasChildViewState 19 Height 20 ID21 IsTrackingViewState 22 **IsValid** 23 NamingContainer 24 Operator 25

ToString

2

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

1

Description

Gets or sets the comparison operation to perform.

Use the **System.Web.UI.WebControls.CompareValidator.Operator** property to specify the comparison operation to perform. The following table lists the comparison operations that are possible.

Page

**Parent** 

PropertiesValid

RenderUplevel

Site

Style

TabIndex

TagKey

TagName

TemplateSourceDirectory

Text

ToolTip

Type

UniqueID

ValueToCompare

**ToString** 

25

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

23

24

25

Gets or sets a constant value to compare with the value entered by the user into the input control being validated.

Use the

 $System. Web. UI. Web Controls. Compare Validator. Value To Compare\ property$ to specify a constant value to compare with the value entered by the user into the input control being validated. If the constant value specified by this property fails to convert to the data type specified by the

System.Web.UI.WebControls.BaseCompareValidator.Type property, a exception is thrown.

ViewState

ViewStateIgnoresCase

Visible

Width

AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

[C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As

HtmlTextWriter)

[JScript] protected override function AddAttributesToRender(writer:

HtmlTextWriter);

n	<b>:</b>	4:
Desc	$r\iota p$	uon

2

3

4

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Adds the attributes of this control to the output stream for rendering on the client. A **System.Web.UI.HtmlTextWriter** that contains the output stream for rendering on the client.

ControlPropertiesValid

[C#] protected override bool ControlPropertiesValid();

[C++] protected: bool ControlPropertiesValid();

[VB] Overrides Protected Function ControlPropertiesValid() As Boolean

[JScript] protected override function ControlPropertiesValid(): Boolean;

### Description

Checks the properties of a the control for valid values.

Return Value: true if the control properties are valid; otherwise false.

**EvaluateIsValid** 

[C#] protected override bool EvaluateIsValid();

[C++] protected: bool EvaluateIsValid();

[VB] Overrides Protected Function EvaluateIsValid() As Boolean

[JScript] protected override function EvaluateIsValid(): Boolean;

## Description

EvaluateIsValid method EvaluateIsValid method

CustomValidator class (System.Web.UI.WebControls)

MS1-863US.APP

	1	Validate
	2	
	3	
	4	Description
	5	Performs user-defined validation on an input control.
	6	Use the System.Web.UI.WebControls.CustomValidator control to
	7	provide a user-defined validation function for an input control. The
	8	System.Web.UI.WebControls.CustomValidator control is separated from the
	9	input control it validates, which allows you to control where the validation
	10	message is displayed.
And the Tark Lens took built that the	11	CustomValidator
M Year Tark	12	Example Syntax:
in the Tan	13	Validate
	14	
ff. Kan fust	15	[C#] public CustomValidator();
	16	[C++] public: CustomValidator();
	17	[VB] Public Sub New()
	18	[JScript] public function CustomValidator();
	19	AccessKey
	20	Attributes
	21	BackColor
	22	BorderColor
	23	BorderStyle
	24	BorderWidth
	25	ChildControlsCreated

ClientID ClientValidationFunction 2 Validate 3 5 Description 6 Gets or sets the name of the custom client-side script function used for 7 validation. Set this property to the name of the function that performs the client-side validation. 10 Context 11 Controls 12 ControlStyle 13 ControlStyleCreated 14 ControlToValidate 15 CssClass 16 Display 17 EnableClientScript 18 Enabled 19 EnableViewState 20 ErrorMessage 21 **Events** 22 Font 23 ForeColor 24 25

HasChildViewState 1683 MS1-863US.APP

	1	Height
	2	ID
	3	IsTrackingViewState
	4	IsValid
	5	NamingContainer
	6	Page
	7	Parent
	8	PropertiesValid
	9	RenderUplevel
	10	Site
AND THE PARTY OF T	11	Style
	12	TabIndex
	13	TagKey
	14	TagName
1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1- 1	15	TemplateSourceDirectory
	16	Text
	17	ToolTip
	18	UniqueID
	19	ViewState
	20	ViewStateIgnoresCase
	21	Visible
	22	Width
	23	Validate
	24	

Description

2

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Occurs when validation is performed on the server.

The System.Web.UI.WebControls.CustomValidator.ServerValidate event is raised when validation is performed on the server. This event is used to provide a custom validation routine for an input control, such as a System.Web.UI.WebControls.TextBox control.

AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

[C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As

HtmlTextWriter)

[JScript] protected override function AddAttributesToRender(writer:

HtmlTextWriter);

Description

Adds the properties of the

System.Web.UI.WebControls.CustomValidator control to the output stream for rendering on the client. A System.Web.UI.HtmlTextWriter that contains the output stream for rendering on the client.

ControlPropertiesValid

[C#] protected override bool ControlPropertiesValid();

Description

25

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Raises the

System.Web.UI.WebControls.CustomValidator.ServerValidate event for the System.Web.UI.WebControls.CustomValidator control.

The System.Web.UI.WebControls.CustomValidator.ServerValidate event is raised when validation is performed on the server. The value to validate.

DataGrid class (System.Web.UI.WebControls)

Validate

#### Description

A data bound list control that displays the items from data source in a table. The **System.Web.UI.WebControls.DataGrid** control allows you to select, sort, and edit these items.

Use the System. Web. UI. Web Controls. Data Grid control to display the

fields of a data source as columns in a table. Each row in the **System.Web.UI.WebControls.DataGrid** control represents a record in the data source. The **System.Web.UI.WebControls.DataGrid** control supports selection, editing, deleting, paging, and sorting.

Validate

[C#] public const string CancelCommandName;

[C++] public: const String\* CancelCommandName;

[VB] Public Const CancelCommandName As String

[JScript] public var CancelCommandName : String;

	:
	1
	1
	1
: =i	1
	1
: dis	1
Control of the contro	1
	1

11	
1	
2	Description
3	Represents the Cancel command name. This field is read-only.
4	Use the
5	System.Web.UI.WebControls.DataGrid.CancelCommandName field to
6	represent the Cancel command name.
7	Validate
8	
9	[C#] public const string DeleteCommandName;
10	[C++] public: const String* DeleteCommandName;
11	[VB] Public Const DeleteCommandName As String
12	[JScript] public var DeleteCommandName : String;
13	
14	Description
15	Represents the <b>Delete</b> command name. This field is read-only.
16	Use the System.Web.UI.WebControls.DataGrid.DeleteCommandName
17	field to represent the <b>Delete</b> command name.
18	Validate
19	
20	[C#] public const string EditCommandName;
21	[C++] public: const String* EditCommandName;
22	[VB] Public Const EditCommandName As String
23	[JScript] public var EditCommandName : String;
24	
25	Description

Represents the **Edit** command name. This field is read-only. 1 Use the System. Web. UI. WebControls. Data Grid. Edit Command Name 2 field to represent the Edit command name. 3 Validate 4 5 [C#] public const string NextPageCommandArgument; 6 [C++] public: const String\* NextPageCommandArgument; 7 [VB] Public Const NextPageCommandArgument As String 8 [JScript] public var NextPageCommandArgument : String; 9 10 Description 11 Represents the **Next** command argument. This field is read-only. 12 Use the 13 System.Web.UI.WebControls.DataGrid.NextPageCommandArgument field 14 to represent the **Next** command argument. 15 Validate 16 17 [C#] public const string PageCommandName; 18 [C++] public: const String\* PageCommandName; 19 [VB] Public Const PageCommandName As String 20 [JScript] public var PageCommandName : String; 21 22 Description 23 Represents the **Page** command name. This field is read-only. 24 25

	1	Use the System.Web.UI.WebControls.DataGrid.PageCommandName
	2	field to represent the Page command name.
	3	Validate
	4	
	5	[C#] public const string PrevPageCommandArgument;
	6	[C++] public: const String* PrevPageCommandArgument;
	7	[VB] Public Const PrevPageCommandArgument As String
	8	[JScript] public var PrevPageCommandArgument : String;
	9	
	10	Description
	11	Represents the Prev command argument. This field is read-only.
	12	Use the
	13	System.Web.UI.WebControls.DataGrid.PrevPageCommandArgument field
U	14	to represent the Prev command argument.
	15	Validate
	16	
	17	[C#] public const string SelectCommandName;
	18	[C++] public: const String* SelectCommandName;
	19	[VB] Public Const SelectCommandName As String
	20	[JScript] public var SelectCommandName : String;
	21	
	22	Description
	23	Represents the <b>Select</b> command name. This field is read-only.

field to represent the **Select** command name.

 $Use the \ {\bf System. Web. UI. WebControls. Data Grid. Select Command Name}$ 

Valida	te

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] public const string SortCommandName;

[C++] public: const String\* SortCommandName;

[VB] Public Const SortCommandName As String

[JScript] public var SortCommandName : String;

Description

Represents the Sort command name. This field is read-only.

Use the **System.Web.UI.WebControls.DataGrid.SortCommandName** field to represent the **Sort** command name.

Validate

[C#] public const string UpdateCommandName;

[C++] public: const String\* UpdateCommandName;

[VB] Public Const UpdateCommandName As String

[JScript] public var UpdateCommandName : String;

Description

Represents the Update command name. This field is read-only.

Use the

System.Web.UI.WebControls.DataGrid.UpdateCommandName field to represent the Update command name.

DataGrid

Example Syntax:

17

18

19

20

21

22

23

24

25

Validate 2 [C#] public DataGrid(); 3 [C++] public: DataGrid(); [VB] Public Sub New() [JScript] public function DataGrid(); 7 Description 8 Initializes a new instance of the System.Web.UI.WebControls.DataGrid 9 class. 10 Use this constructor to create and initialize a new instance of the 11 System.Web.UI.WebControls.DataGrid class. 12 AccessKey 13 AllowCustomPaging 14 Validate 15

Description

Gets or sets a value that indicates whether custom paging is enabled.

Paging allows you to display the contents of the

System.Web.UI.WebControls.DataGrid control in page segments. The number of items on a page is determined by the

System.Web.UI.WebControls.DataGrid.PageSize property. If no value is specified for the System.Web.UI.WebControls.DataGrid.PageSize property, the System.Web.UI.WebControls.DataGrid will display 10 items on a page.

1	AllowPaging
2	Validate
3	
4	[C#] public virtual bool AllowPaging {get; set;}
5	[C++] public:property virtual bool get_AllowPaging();public:property
6	virtual void set_AllowPaging(bool);
7	[VB] Overridable Public Property AllowPaging As Boolean
8	[JScript] public function get AllowPaging(): Boolean;public function set
9	AllowPaging(Boolean);
10	
11	Description
12	Gets or sets a value that indicates whether paging is enabled.
13	Paging allows you to display the contents of the
14	System.Web.UI.WebControls.DataGrid control in page segments. The number
15	of items on a page is determined by the
16	System.Web.UI.WebControls.DataGrid.PageSize property. If no value is
17	specified for the System.Web.UI.WebControls.DataGrid.PageSize property, the
18	System.Web.UI.WebControls.DataGrid control will display 10 items on a page.
19	AllowSorting
20	Validate
21	
22	[C#] public virtual bool AllowSorting {get; set;}
23	[C++] public:property virtual bool get_AllowSorting();public:property
24	virtual void set_AllowSorting(bool);
25	[VB] Overridable Public Property AllowSorting As Boolean

[JScript] public function get AllowSorting(): Boolean; public function set AllowSorting(Boolean);

Description

Gets or sets a value that indicates whether sorting is enabled.

When sorting is enabled, System.Web.UI.WebControls.LinkButton controls are rendered in the heading section of each column where the System.Web.UI.WebControls.DataGridColumn.SortExpression property is set. These System.Web.UI.WebControls.LinkButton controls allow you to sort the System.Web.UI.WebControls.DataGrid control by the selected column. The only exception is when you use a

System.Web.UI.WebControls.TemplateColumn column type with the System.Web.UI.WebControls.TemplateColumn.HeaderTemplate property set. In this case, you must provide a System.Web.UI.WebControls.Button control in the System.Web.UI.WebControls.TemplateColumn.HeaderTemplate of the column.

AlternatingItemStyle

Validate

[C#] public virtual TableItemStyle AlternatingItemStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get AlternatingItemStyle();

[VB] Overridable Public ReadOnly Property AlternatingItemStyle As

**TableItemStyle** 

[JScript] public function get AlternatingItemStyle(): TableItemStyle;

lee@hayes ρIIc 509-324-9256 1694 MS1-863US.APP

21

19

20

23

22

24

25

Description

2

3

5

8

9

12

13

14

Gets the style properties for alternating items in the System.Web.UI.WebControls.DataGrid control.

Use the System. Web. UI. Web Controls. Data Grid. Alternating Item Style property to provide a custom style for the alternating items in the System.Web.UI.WebControls.DataGrid control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the System.Web.UI.WebControls.DataGrid control.

Attributes

**AutoGenerateColumns** 

Validate

Description

Gets or sets a value that indicates whether

System.Web.UI.WebControls.BoundColumn objects are automatically created and displayed in the System. Web. UI. Web Controls. Data Grid control for each field in the data source.

Use this property to automatically create a

System.Web.UI.WebControls.BoundColumn object for each field in the data source. Each field is then rendered as a column in the

System.Web.UI.WebControls.DataGrid control in the order that the fields appear in the data source.

BackColor 1 BackImageUrl Validate Description Gets or sets the URL of an image to display in the background of the System.Web.UI.WebControls.DataGrid control. Use the System.Web.UI.WebControls.DataGrid.BackImageUrl property to specify an image to display in the background of the System.Web.UI.WebControls.DataGrid control. BorderColor BorderStyle BorderWidth CellPadding CellSpacing ChildControlsCreated ClientID Columns Validate 21 22 Description 23 Gets a collection of objects that represent the columns of the 24 System.Web.UI.WebControls.DataGrid control.

MS1-863US.APP

5

6

7

8

9

10

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use this property to programmatically control the collection of columns in the System.Web.UI.WebControls.DataGrid control. The

System.Web.UI.WebControls.DataGrid.Columns collection contains explicitly declared columns that get rendered in the

System.Web.UI.WebControls.DataGrid control.

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

CurrentPageIndex

Validate

#### Description

Gets or sets the index of the currently displayed page.

Use this property to determine the currently displayed page in the System.Web.UI.WebControls.DataGrid control when paging is enabled. This property is also used to programmatically control which page is displayed.

DataKeyField

DataKeys

DataKeysArray

DataMember

DataSource

EditItemIndex

19

20

21

22

23

24

25

Validate

Description

2

3

5

7

9

Gets or sets the index of an item in the

System.Web.UI.WebControls.DataGrid control to edit.

When the System. Web. UI. Web Controls. Data Grid control contains an System.Web.UI.WebControls.EditCommandColumn object, use this property to determine the index of the item selected in the System.Web.UI.WebControls.DataGrid control to edit.

EditItemStyle

Validate

[C#] public virtual TableItemStyle EditItemStyle {get;}

[C++] public: property virtual TableItemStyle\* get EditItemStyle();

[VB] Overridable Public ReadOnly Property EditItemStyle As TableItemStyle

[JScript] public function get EditItemStyle(): TableItemStyle;

Description

Gets the style properties of the item selected for editing in the System.Web.UI.WebControls.DataGrid control.

Use the **System.Web.UI.WebControls.DataGrid.EditItemStyle** property to provide a custom style for the item selected for editing in the System.Web.UI.WebControls.DataGrid control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within

the cell. Providing a different style enhances the appearance of the System.Web.UI.WebControls.DataGrid control. 2 Enabled 3 **EnableViewState Events** 5 Font FooterStyle Validate 8 10 Description Gets the style properties of the footer section in the 12 System.Web.UI.WebControls.DataGrid control. 13 Use this property to provide a custom style for the footer of the 14 System.Web.UI.WebControls.DataGrid control. Common style attributes that 15 can be adjusted include forecolor, backcolor, font, and content alignment within 16 the cell. Providing a different style enhances the appearance of the 17 System.Web.UI.WebControls.DataGrid control. 18 ForeColor 19 GridLines 20 HasChildViewState 21 HeaderStyle 22 Validate 23

24

25

16

17

18

19

20

21

22

23

24

25

# Description

2

3

5

6

9

Gets the style properties of the heading section in the System.Web.UI.WebControls.DataGrid control.

Use this property to provide a custom style for the heading section of the System.Web.UI.WebControls.DataGrid control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the System.Web.UI.WebControls.DataGrid control.

Height

HorizontalAlign

ID

IsTrackingViewState

Items

Validate

#### Description

Gets a collection of System. Web.UI. WebControls. Data GridItem objects that represent the individual items in the

System.Web.UI.WebControls.DataGrid control.

Use the System. Web. UI. Web Controls. Data Grid. Items collection to programmatically control the items in the

System.Web.UI.WebControls.DataGrid control. The

3

4

5

6

7

8

9

12

13

14

16

17

18

19

20

21

22

23

24

25

System.Web.UI.WebControls.DataGrid.Items collection does not provide any methods to add or remove items to the collection. However, you can control the contents of an item by providing a handler for the

System.Web.UI.WebControls.DataGrid.ItemCreated event.

ItemStyle

Validate

[C#] public virtual TableItemStyle ItemStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get\_ItemStyle();

[VB] Overridable Public ReadOnly Property ItemStyle As TableItemStyle

[JScript] public function get ItemStyle(): TableItemStyle;

Description

Gets the style properties of the items in the

System.Web.UI.WebControls.DataGrid control.

System.Web.UI.WebControls.DataGrid control.

Use this property to provide a custom style for the items of the System.Web.UI.WebControls.DataGrid control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the

NamingContainer

Page

**PageCount** 

Validate

1701 MS1-863US.APP lee@hayes plic 509+324+9256

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Description

Gets the total number of pages required to display the items in the System. Web.UI. WebControls. DataGrid control.

Use this property to programmatically determine the number of pages required to display the items in the **System.Web.UI.WebControls.DataGrid** control. This property is only used when the

 ${\bf System. Web. UI. WebControls. Data Grid. Allow Paging\ property\ is\ set\ to\ true\ .}$ 

PagerStyle

Validate

[C#] public virtual DataGridPagerStyle PagerStyle {get;}

[C++] public: \_\_property virtual DataGridPagerStyle\* get\_PagerStyle();

[VB] Overridable Public ReadOnly Property PagerStyle As DataGridPagerStyle

[JScript] public function get PagerStyle() : DataGridPagerStyle;

Description

Gets the style properties of the paging section of the System.Web.UI.WebControls.DataGrid control.

Use this property to provide a custom style for the paging section of the **System.Web.UI.WebControls.DataGrid** control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the

 ${\bf System. Web. UI. WebControls. Data Grid\ control.}$ 

1	PageSize
2	Validate
3	
4	[C#] public virtual int PageSize {get; set;}
5	[C++] public:property virtual int get_PageSize();public:property virtual void
6	set_PageSize(int);
7	[VB] Overridable Public Property PageSize As Integer
8	[JScript] public function get PageSize(): int;public function set PageSize(int);
9	
10	Description
11	Gets or sets the number of items to display on a single page of the
12	System.Web.UI.WebControls.DataGrid control.
13	Use this property to specify the number of items to display on a single page
14	of the System.Web.UI.WebControls.DataGrid control. The
15	System.Web.UI.WebControls.DataGrid.AllowPaging property must be set to
16	true for this property have any effect.
17	Parent
18	SelectedIndex
19	Validate
20	
21	
22	Description
23	Gets or sets the index of the selected item in the
24	System.Web.UI.WebControls.DataGrid control.
25	

Use the System.Web.UI.WebControls.DataGrid.SelectedIndex property to determine the index of the item selected by the user in the System.Web.UI.WebControls.DataGrid control. You can also use this property to programmatically specify which item is selected in the System.Web.UI.WebControls.DataGrid control.

SelectedItem

Validate

[C#] public virtual DataGridItem SelectedItem {get;}

[C++] public: \_\_property virtual DataGridItem\* get\_SelectedItem();

[VB] Overridable Public ReadOnly Property SelectedItem As DataGridItem

[JScript] public function get SelectedItem(): DataGridItem;

Description

Gets a **System.Web.UI.WebControls.DataGridItem** object that represents the selected item in the **System.Web.UI.WebControls.DataGrid** control.

Use the System.Web.UI.WebControls.DataGrid.SelectedItem property to get a System.Web.UI.WebControls.DataGridItem object that represents the selected item in the System.Web.UI.WebControls.DataGrid control. This object can then be used to access the properties of the selected item.

SelectedItemStyle

Validate

 $[C\#]\ public\ virtual\ Table I tem Style\ Selected I tem Style\ \{get;\}$ 

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C++] public: \_\_property virtual TableItemStyle\* get\_SelectedItemStyle(); [VB] Overridable Public ReadOnly Property SelectedItemStyle As TableItemStyle [JScript] public function get SelectedItemStyle(): TableItemStyle; Description Gets the style properties of the currently selected item in the System.Web.UI.WebControls.DataGrid control.  $Use\ the\ System. Web. UI. WebControls. Data Grid. Selected Item Style$ property to provide a custom style for the selected item in the System.Web.UI.WebControls.DataGrid control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the  ${\bf System. Web. UI. WebControls. Data Grid\ control.}$ ShowFooter Validate [C#] public virtual bool ShowFooter {get; set;} [C++] public: \_\_property virtual bool get\_ShowFooter();public: \_\_property virtual void set ShowFooter(bool); [VB] Overridable Public Property ShowFooter As Boolean [JScript] public function get ShowFooter(): Boolean; public function set ShowFooter(Boolean); Description

Gets or sets a value that indicates whether the footer is displayed in the System.Web.UI.WebControls.DataGrid control.

Set this property to true to display the footer in the

System.Web.UI.WebControls.DataGrid control. The appearance of the footer is controlled by using the System.Web.UI.WebControls.DataGrid.FooterStyle property.

ShowHeader

Validate

[C#] public virtual bool ShowHeader {get; set;}

[C++] public: \_\_property virtual bool get\_ShowHeader();public: \_\_property virtual void set\_ShowHeader(bool);

[VB] Overridable Public Property ShowHeader As Boolean

[JScript] public function get ShowHeader() : Boolean; public function set ShowHeader(Boolean);

Description

Gets or sets a value that indicates whether the header is displayed in the System.Web.UI.WebControls.DataGrid control.

Set this property to **true** to display the header in the **System.Web.UI.WebControls.DataGrid**. The appearance of the header is controlled by using the **System.Web.UI.WebControls.DataGrid.HeaderStyle** property.

Site

Style

TabIndex TagKey 2 TagName 3 **TemplateSourceDirectory** ToolTip 5 UniqueID ViewState ViewStateIgnoresCase 8 VirtualItemCount 9 Validate 10 11 12 Description 13 Gets or sets the virtual number of items in the 14 System.Web.UI.WebControls.DataGrid control when custom paging is used. 15 Use this property to specify the virtual number of items in the 16  ${\bf System. Web. UI. WebControls. Data Grid\ control\ when\ custom\ paging\ is\ used.}$ 17 This property is only used when the 18 System.Web.UI.WebControls.DataGrid.AllowCustomPaging property is set to 19 true. 20 Visible 21 Width 22 Validate 23 24 25

Description

Occurs when the **Cancel** button is clicked for an item in the **System.Web.UI.WebControls.DataGrid** control.

The System.Web.UI.WebControls.DataGrid.CancelCommand event is raised when the Cancel button is clicked for an item in the System.Web.UI.WebControls.DataGrid control.

Validate

Description

Occurs when the **Delete** button is clicked for an item in the **System.Web.UI.WebControls.DataGrid** control.

The System.Web.UI.WebControls.DataGrid.DeleteCommand event is raised when the Delete button is clicked for an item in the System.Web.UI.WebControls.DataGrid control.

Validate

Description

Occurs when the **Edit** button is clicked for an item in the **System.Web.UI.WebControls.DataGrid** control.

5

6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

The System.Web.UI.WebControls.DataGrid.EditCommand event is raised when the Edit button is clicked for an item in the System.Web.UI.WebControls.DataGrid control.

Validate

### Description

Occurs when any button is clicked in the System.Web.UI.WebControls.DataGrid control.

The System.Web.UI.WebControls.DataGrid.ItemCommand event is raised when any button is clicked in the System.Web.UI.WebControls.DataGrid control. This event is commonly used to handle buttons controls with a custom CommandName value in the System.Web.UI.WebControls.DataGrid control.

Validate

[C#] public event DataGridItemEventHandler ItemCreated;

[C++] public: \_\_event DataGridItemEventHandler\* ItemCreated;

[VB] Public Event ItemCreated As DataGridItemEventHandler

### Description

Occurs on the server when an item in the

System.Web.UI.WebControls.DataGrid control is created.

The System.Web.UI.WebControls.DataGrid.ItemCreated event is raised when an item in the System.Web.UI.WebControls.DataGrid control is created, both during round-trips and at data bind time.

ij
ij
ïIJ
(9
#
; san
'[
ı çələ
t ingit

•	١
2	
3	
4	
5	
,	
O	'
7	
8	
9	
10	,
11	
12	
13	
14	
15	
16	
17	
10	
10	'
19	'
20	
21	١
22	
23	
24	

1	7.	. 1	:	4	_	te
١,	12	11	1	П	Я	TE

[C#] public event DataGridItemEventHandler ItemDataBound;

[C++] public: event DataGridItemEventHandler\* ItemDataBound;

[VB] Public Event ItemDataBound As DataGridItemEventHandler

### Description

Occurs after an item is data bound to the

## System.Web.UI.WebControls.DataGrid control.

The System.Web.UI.WebControls.DataGrid.ItemDataBound event is raised after an item is data bound to the System.Web.UI.WebControls.DataGrid control. This event provides you with the last opportunity to access the data item before it is displayed on the client. After this event is raised, the data item is nulled out and no longer available.

Validate

### Description

Occurs when one of the page selection elements is clicked.

The System.Web.UI.WebControls.DataGrid.PageIndexChanged event is raised when one of the page selection elements is clicked.

Validate

Description

Occurs when a column is sorted.

The **System.Web.UI.WebControls.DataGrid.SortCommand** event is raised when a column is sorted.

Validate

Description

2

4

5

6

8

10

12

13

14

15

16

17

18

19

20

21

22

23

24

Occurs when the **Update** button is clicked for an item in the **System.Web.UI.WebControls.DataGrid** control.

The **System.Web.UI.WebControls.DataGrid.UpdateCommand** event is raised when the **Update** button for an item is clicked.

CreateColumnSet

[C#] protected virtual ArrayList CreateColumnSet(PagedDataSource dataSource, bool useDataSource);

[C++] protected: virtual ArrayList\* CreateColumnSet(PagedDataSource\* dataSource, bool useDataSource);

[VB] Overridable Protected Function CreateColumnSet(ByVal dataSource As PagedDataSource, ByVal useDataSource As Boolean) As ArrayList

 $[JScript]\ protected\ function\ CreateColumnSet(dataSource: PagedDataSource,$ 

useDataSource: Boolean): ArrayList;

Description

Creates the set of columns to be used to build up the control hierarchy.

When AutoGenerateColumns is true, the columns are created to match the

3

5

9

10

11

12

13

14

15

16

17

18

19

20

22

24

25

datasource and are appended to the set of columns defined in the Columns collection. The datasource being used to create the control hierarchy Whether to use the datasource to generate columns automatically or to use saved state.

CreateControlHierarchy

[C#] protected override void CreateControlHierarchy(bool useDataSource);

[C++] protected: void CreateControlHierarchy(bool useDataSource);

[VB] Overrides Protected Sub CreateControlHierarchy(ByVal useDataSource As Boolean)

[JScript] protected override function CreateControlHierarchy(useDataSource : Boolean);

### Description

Creates the control hierarchy that is used to render the DataGrid. This is called whenever a control hierarchy is needed and the ChildControlsCreated property is false. The implementation assumes that all the children in the controls collection have already been cleared. Whether to use the datasource to generate columns automatically or to use saved state.

### CreateControlStyle

[C#] protected override Style CreateControlStyle();

[C++] protected: Style\* CreateControlStyle();

[VB] Overrides Protected Function CreateControlStyle() As Style

[JScript] protected override function CreateControlStyle(): Style;

lee ⊗hayes pilc 509+324-9256 1712 MS1-863US.APP

Description 2 Creates new control style. 3 Return Value: A System.Web.UI.WebControls.Style the represents the new style. 5 CreateItem 6 7 [C#] protected virtual DataGridItem CreateItem(int itemIndex, int 8 dataSourceIndex, ListItemType itemType); [C++] protected: virtual DataGridItem\* CreateItem(int itemIndex, int 10 dataSourceIndex, ListItemType itemType); 11 [VB] Overridable Protected Function CreateItem(ByVal itemIndex As Integer, 12 ByVal dataSourceIndex As Integer, ByVal itemType As ListItemType) As 13 DataGridItem 14 [JScript] protected function CreateItem(itemIndex: int, dataSourceIndex: int, 15 itemType: ListItemType): DataGridItem; 16 17 Description 18 19 InitializeItem 20 21 [C#] protected virtual void InitializeItem(DataGridItem item, DataGridColumn[] 22 columns); 23 [C++] protected: virtual void InitializeItem(DataGridItem\* item, 24

DataGridColumn\* columns[]);

24

1	[VB] Overridable Protected Sub InitializeItem(ByVal item As DataGridItem,
2	ByVal columns() As DataGridColumn)
3	[JScript] protected function InitializeItem(item : DataGridItem, columns :
4	DataGridColumn[]);
5	
6	Description
7	
8	InitializePager
9	
10	[C#] protected virtual void InitializePager(DataGridItem item, int columnSpan,
11	PagedDataSource pagedDataSource);
12	[C++] protected: virtual void InitializePager(DataGridItem* item, int columnSpan,
13	PagedDataSource* pagedDataSource);
14	[VB] Overridable Protected Sub InitializePager(ByVal item As DataGridItem,
15	ByVal columnSpan As Integer, ByVal pagedDataSource As PagedDataSource)
16	[JScript] protected function InitializePager(item : DataGridItem, columnSpan : int,
17	pagedDataSource : PagedDataSource);
18	
19	Description
20	Creates a DataGridItem that contains the paging UI. The paging UI is a
21	navigation bar that is a built into a single TableCell that spans across all columns
22	of the DataGrid.

LoadViewState

[C#] protected override void LoadViewState(object savedState);

1	[C++] protected: void LoadViewState(Object* savedState);
2	[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)
3	[JScript] protected override function LoadViewState(savedState : Object);
4	
5	Description
6	Loads a saved state of the System.Web.UI.WebControls.DataGrid . A
7	saved state of the System.Web.UI.WebControls.DataGrid.
8	OnBubbleEvent
9	
10	[C#] protected override bool OnBubbleEvent(object source, EventArgs e);
11	[C++] protected: bool OnBubbleEvent(Object* source, EventArgs* e);
12	[VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,
13	ByVal e As EventArgs) As Boolean
14	[JScript] protected override function OnBubbleEvent(source : Object, e :
15	EventArgs): Boolean;
16	
17	Description
18	The source of the event. An System.EventArgs that contains event data.
19	OnCancelCommand
20	
21	[C#] protected virtual void OnCancelCommand(DataGridCommandEventArgs e)
22	[C++] protected: virtual void OnCancelCommand(DataGridCommandEventArgs
23	e);
24	[VB] Overridable Protected Sub OnCancelCommand(ByVal e As
25	DataGridCommandEventArgs)

[JScript] protected function OnCancelCommand(e: DataGridCommandEventArgs); 2 3 Description Raises the System. Web. UI. Web Controls. Data Grid. Cancel Command 5 event. This allows you to provide a custom handler for the event. Use the 7 System.Web.UI.WebControls.DataGrid.OnCancelCommand(System.Web.UI 8 .WebControls.DataGridCommandEventArgs) method to provide a custom handler for the System. Web. UI. WebControls. Data Grid. Cancel Command 10 event. A System. Web. UI. Web Controls. Data Grid Command Event Args that 11 contains event data. 12 **OnDeleteCommand** 13 14 [C#] protected virtual void OnDeleteCommand(DataGridCommandEventArgs e); 15 [C++] protected: virtual void OnDeleteCommand(DataGridCommandEventArgs\* e); 17 [VB] Overridable Protected Sub OnDeleteCommand(ByVal e As 18 DataGridCommandEventArgs) 19 [JScript] protected function OnDeleteCommand(e: 20 DataGridCommandEventArgs); 21 22 Description 23 Raises the System. Web. UI. Web Controls. Data Grid. Delete Command 24 event. This allows you to provide a custom handler for the event.

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use the

System.Web.UI.WebControls.DataGrid.OnDeleteCommand(System.Web.UI. WebControls.DataGridCommandEventArgs) method to provide a custom handler for the System. Web. UI. Web Controls. Data Grid. Delete Command event. A System. Web. UI. WebControls. Data Grid Command Event Args that contains event data.

**OnEditCommand** 

[C#] protected virtual void OnEditCommand(DataGridCommandEventArgs e); [C++] protected: virtual void OnEditCommand(DataGridCommandEventArgs\* e);

[VB] Overridable Protected Sub OnEditCommand(ByVal e As

DataGridCommandEventArgs)

[JScript] protected function OnEditCommand(e: DataGridCommandEventArgs);

Description

Raises the System. Web. UI. Web Controls. Data Grid. Edit Command event. This allows you to provide a custom handler for the event.

Use the

System.Web.UI.WebControls.DataGrid.OnEditCommand(System.Web.UI.W ebControls.DataGridCommandEventArgs) method to provide a custom handler for the System.Web.UI.WebControls.DataGrid.EditCommand event. A System.Web.UI.WebControls.DataGridCommandEventArgs that contains event data.

1717

OnItemCommand

1	
2	[C#] protected virtual void OnItemCommand(DataGridCommandEventArgs e);
3	[C++] protected: virtual void OnItemCommand(DataGridCommandEventArgs*
4	e);
5	[VB] Overridable Protected Sub OnItemCommand(ByVal e As
6	DataGridCommandEventArgs)
7	[JScript] protected function OnItemCommand(e: DataGridCommandEventArgs)
8	
9	Description
10	Raises the System.Web.UI.WebControls.DataGrid.ItemCommand
11	event. This allows you to provide a custom handler for the event. This allows you
12	to provide a custom handler for the event.
13	Use the
14	System. We b. UI. We bC ontrols. Data Grid. On Item Command (System. Web. UI. WebControls.) A significant control of the con
15	WebControls.DataGridCommandEventArgs) method to provide a custom
16	handler for the System.Web.UI.WebControls.DataGrid.ItemCommand event.
17	A System.Web.UI.WebControls.DataGridCommandEventArgs that contains
18	event data.
19	OnItemCreated
20	
21	[C#] protected virtual void OnItemCreated(DataGridItemEventArgs e);
22	[C++] protected: virtual void OnItemCreated(DataGridItemEventArgs* e);
23	[VB] Overridable Protected Sub OnItemCreated(ByVal e As
24	DataGridItemEventArgs)
25	[JScript] protected function OnItemCreated(e: DataGridItemEventArgs);

•			

Descr	rinti	on
Desci	pu	On

3

5

6

8

9

10

11

12

14

15

16

17

18

19

20

21

22

23

24

Raises the **System.Web.UI.WebControls.DataGrid.ItemCreated** event.

This allows you to provide a custom handler for the event.

Use the

System.Web.UI.WebControls.DataGrid.OnItemCreated(System.Web.UI.WebControls.DataGridItemEventArgs) method to provide a custom handler for the System.Web.UI.WebControls.DataGrid.ItemCreated event. A System.Web.UI.WebControls.DataGridItemEventArgs that contains event data.

**OnItemDataBound** 

[C#] protected virtual void OnItemDataBound(DataGridItemEventArgs e);

[C++] protected: virtual void OnItemDataBound(DataGridItemEventArgs\* e);

[VB] Overridable Protected Sub OnItemDataBound(ByVal e As

DataGridItemEventArgs)

[JScript] protected function OnItemDataBound(e: DataGridItemEventArgs);

Description

Raises the **System.Web.UI.WebControls.DataGrid.ItemDataBound** event. This allows you to provide a custom handler for the event.

Use the

System.Web.UI.WebControls.DataGrid.OnItemDataBound(System.Web.UI. WebControls.DataGridItemEventArgs) method to provide a custom handler for the System.Web.UI.WebControls.DataGrid.ItemDataBound event. A

	1	System.Web.UI.WebControls.DataGridItemEventArgs that contains event
	2	data.
	3	OnPageIndexChanged
	4	
	5	[C#] protected virtual void
	6	OnPageIndexChanged(DataGridPageChangedEventArgs e);
	7	[C++] protected: virtual void
	8	OnPageIndexChanged(DataGridPageChangedEventArgs* e);
2	9	[VB] Overridable Protected Sub OnPageIndexChanged(ByVal e As
ind frait this	10	DataGridPageChangedEventArgs)
na fina	11	[JScript] protected function OnPageIndexChanged(e:
11: Juny ,011	12	DataGridPageChangedEventArgs);
7	13	
	14	Description
;	15	Raises the System.Web.UI.WebControls.DataGrid.PageIndexChanged
	16	event. This allows you to provide a custom handler for the event.
	17	Use the
	18	System.Web.UI.WebControls.DataGrid.OnPageIndexChanged(System.Web.
	19	UI.WebControls.DataGridPageChangedEventArgs) method to provide a
	20	custom handler for the
	21	System.Web.UI.WebControls.DataGrid.PageIndexChanged event. A
	22	System.Web.UI.WebControls.DataGridPageChangedEventArgs that contains
	23	event data.
	24	OnSortCommand
	25	

П	
1	
2	[C#] protected virtual void OnSortCommand(DataGridSortCommandEventArgs
3	e);
4	[C++] protected: virtual void
5	OnSortCommand(DataGridSortCommandEventArgs* e);
6	[VB] Overridable Protected Sub OnSortCommand(ByVal e As
7	DataGridSortCommandEventArgs)
8	[JScript] protected function OnSortCommand(e:
9	DataGridSortCommandEventArgs);
10	
11	Description
12	Raises the System.Web.UI.WebControls.DataGrid.SortCommand
13	event. This allows you to provide a custom handler for the event.
14	Use the
	System Web III WebControls DataGrid OnSortCommand(System.Web.UI

System.Web.UI.WebControls.DataGrid.OnSortCommand(System.Web.UI.WebControls.DataGridSortCommandEventArgs) method to provide a custom handler for the System.Web.UI.WebControls.DataGrid.SortCommand event.

A System.Web.UI.WebControls.DataGridSortCommandEventArgs that contains event data.

OnUpdateCommand

[C#] protected virtual void OnUpdateCommand(DataGridCommandEventArgs e);
[C++] protected: virtual void OnUpdateCommand(DataGridCommandEventArgs\*
e);

[VB] Overridable Protected Sub OnUpdateCommand(ByVal e As

1	DataGridCommandEventArgs)
2	[JScript] protected function OnUpdateCommand(e:
3	DataGridCommandEventArgs);
4	
5	Description
6	Raises the System.Web.UI.WebControls.DataGrid.UpdateCommand
7	event. This allows you to provide a custom handler for the event.
8	Use the
9	System. Web. UI. Web Controls. Data Grid. On Update Command (System. Web. UI.) A state of the control of the
10	.WebControls.DataGridCommandEventArgs) method to provide a custom
11	handler for the System.Web.UI.WebControls.DataGrid.UpdateCommand
12	event. A System.Web.UI.WebControls.DataGridCommandEventArgs that
13	contains event data.
14	PrepareControlHierarchy
15	
16	[C#] protected override void PrepareControlHierarchy();
17	[C++] protected: void PrepareControlHierarchy();
18	[VB] Overrides Protected Sub PrepareControlHierarchy()
19	[JScript] protected override function PrepareControlHierarchy();
20	
21	Description
22	SaveViewState
23	
24	[C#] protected override object SaveViewState();
25	[C++] protected: Object* SaveViewState();

MS1-863US.APP

1	[VB] Overrides Protected Function SaveViewState() As Object
2	[JScript] protected override function SaveViewState() : Object;
3	
4	Description
5	Saves the current state of the System.Web.UI.WebControls.DataGrid.
6	Return Value: The saved state of the System.Web.UI.WebControls.DataGrid.
7	TrackViewState
8	
9	[C#] protected override void TrackViewState();
10	[C++] protected: void TrackViewState();
11	[VB] Overrides Protected Sub TrackViewState()
12	[JScript] protected override function TrackViewState();
13	
14	Description
15	Marks the starting point to begin tracking and saving changes to the control
16	as part of the control viewstate.
17	DataGridColumn class (System.Web.UI.WebControls)
18	TrackViewState
19	
20	
21	Description
22	Serves as the base class for the different column types of the
23	System.Web.UI.WebControls.DataGrid control.
24	The System.Web.UI.WebControls.DataGridColumn class is the base
25	class for all column types of the System.Web.UI.WebControls.DataGrid

1	control. It defines the properties and methods that are common to all column
2	types. An instance of the System.Web.UI.WebControls.DataGridColumn class
3	is typically not created directly.
4	DataGridColumn
5	Example Syntax:
6	TrackViewState
7	
8	[C#] public DataGridColumn();
9	[C++] public: DataGridColumn();
10	[VB] Public Sub New()
11	[JScript] public function DataGridColumn();
12	
13	Description
14	Initializes a new instance of the
15	System.Web.UI.WebControls.DataGridColumn class.
16	Use this constructor to create and initialize an instance of the
17	System.Web.UI.WebControls.DataGridColumn class.
18	DesignMode
19	TrackViewState
20	
21	[C#] protected bool DesignMode {get;}
22	[C++] protected:property bool get_DesignMode();
23	[VB] Protected ReadOnly Property DesignMode As Boolean
24	[JScript] protected function get DesignMode(): Boolean;
25	

2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	l

$\mathbf{r}$				
1)2	esci	nn	111	าท
$\boldsymbol{\mathcal{L}}$	ノリレル	$\iota \nu$	$\iota\iota\iota$	,,,

Gets a value that indicates whether the column is in design mode.

Use the **System.Web.UI.WebControls.DataGridColumn.DesignMode** property to programmatically determine whether the column is in design mode

FooterStyle

**TrackViewState** 

[C#] public virtual TableItemStyle FooterStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get FooterStyle();

[VB] Overridable Public ReadOnly Property FooterStyle As TableItemStyle

[JScript] public function get FooterStyle() : TableItemStyle;

### Description

Gets the style properties for the footer section of the column.

Use this property to provide a custom style for the footer section of the column. Common style attributes that can be adjusted, include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the column in the

# System.Web.UI.WebControls.DataGrid control.

FooterText

**TrackViewState** 

[C#] public virtual string FooterText {get; set;}

[C++] public: \_\_property virtual String\* get FooterText();public: \_\_property

1	virtual void set_FooterText(String*);
2	[VB] Overridable Public Property FooterText As String
3	[JScript] public function get FooterText(): String;public function set
4	FooterText(String);
5	
6	Description
7	Gets or sets the text displayed in the footer section of the column.
8	Use the System.Web.UI.WebControls.DataGridColumn.FooterText
9	property to specify or determine the text displayed in the footer section of the
10	column.
11	HeaderImageUrl
12	TrackViewState
13	
14	[C#] public virtual string HeaderImageUrl {get; set;}
15	[C++] public:property virtual String* get_HeaderImageUrl();public:property
16	virtual void set_HeaderImageUrl(String*);
17	[VB] Overridable Public Property HeaderImageUrl As String
18	[JScript] public function get HeaderImageUrl(): String;public function set
19	HeaderImageUrl(String);
20	
21	Description
22	Gets or sets the location of an image to display in the header section of the
23	column.
24	Use the
25	System.Web.UI.WebControls.DataGridColumn.HeaderImageUrl property to

specify the URL of an image to display in the header section of the column. You can use a relative or an absolute URL. A relative URL relates the location of the image to the location of the Web page without specifying a complete path on the server. The path is relative to the location of the Web page. This makes it easier to move the entire site to another directory on the server without updating the path to the image in code. An absolute URL provides the complete path, so moving the site to another directory requires updating the code.

HeaderStyle

TrackViewState

[C#] public virtual TableItemStyle HeaderStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get\_HeaderStyle();

[VB] Overridable Public ReadOnly Property HeaderStyle As TableItemStyle

[JScript] public function get HeaderStyle(): TableItemStyle;

### Description

Gets the style properties for the header section of the column.

Use this property to provide a custom style for the header section of the column. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the column in the

## System.Web.UI.WebControls.DataGrid control.

HeaderText

TrackViewState

1	
2	[C#] public virtual string HeaderText {get; set;}
3	[C++] public:property virtual String* get_HeaderText();public:property
4	virtual void set_HeaderText(String*);
5	[VB] Overridable Public Property HeaderText As String
6	[JScript] public function get HeaderText(): String; public function set
7	HeaderText(String);
8	
9	Description
10	Gets or sets the text displayed in the header section of the column.
11	Use the System.Web.UI.WebControls.DataGridColumn.HeaderText
12	property to specify or determine the text displayed in the footer section of the
13	column.
14	IsTrackingViewState
15	TrackViewState
16	
17	[C#] protected bool IsTrackingViewState {get;}
18	[C++] protected:property bool get_IsTrackingViewState();
19	[VB] Protected ReadOnly Property IsTrackingViewState As Boolean
20	[JScript] protected function get IsTrackingViewState(): Boolean;
21	
22	Description
23	Determines if the System.Web.UI.WebControls.DataGridColumn is
24	marked to save its state.
25	ItemStyle

1	TrackViewState
2	
3	[C#] public virtual TableItemStyle ItemStyle {get;}
4	[C++] public:property virtual TableItemStyle* get_ItemStyle();
5	[VB] Overridable Public ReadOnly Property ItemStyle As TableItemStyle
6	[JScript] public function get ItemStyle(): TableItemStyle;
7	
8	Description
9	Gets the style properties for the item cells of the column.
10	Use this property to provide a custom style for the item cells of the column
11	Common style attributes that can be adjusted include forecolor, backcolor, font,
12	and content alignment within the cell. Providing a different style enhances the
13	appearance of the column in the System.Web.UI.WebControls.DataGrid
14	control.
15	Owner
16	TrackViewState
17	
18	[C#] protected DataGrid Owner {get;}
19	[C++] protected:property DataGrid* get_Owner();
20	[VB] Protected ReadOnly Property Owner As DataGrid
21	[JScript] protected function get Owner(): DataGrid;

Description

23

24

Gets the **System.Web.UI.WebControls.DataGrid** control that the column is a member of.

Use the System. Web. UI. WebControls. Data Grid Column. Owner 1 property to programmatically determine the 2 System. Web. UI. Web Controls. Data Grid control that the column is a member of. 3 SortExpression **TrackViewState** 5 6 [C#] public virtual string SortExpression {get; set;} 7 [C++] public: property virtual String\* get SortExpression(); public: property 8 virtual void set SortExpression(String\*); 9 [VB] Overridable Public Property SortExpression As String 10 [JScript] public function get SortExpression(): String; public function set 11 SortExpression(String); 12 13 Description 14 Gets or sets the name of the field to pass to the 15 System. Web. UI. WebControls. Data Grid. On Sort Command (System. Web. UI. WebControls.)16 ebControls.DataGridSortCommandEventArgs) method when a column is 17 selected for sorting. 18 Use the 19 System.Web.UI.WebControls.DataGridColumn.SortExpression property to 20 specify or determine the name of the field to pass to the 21 System.Web.UI.WebControls.DataGrid.OnSortCommand(System.Web.UI.W 22 ebControls.DataGridSortCommandEventArgs) method when a column is 23 selected for sorting. 24 ViewState 25

1	TrackViewState
2	
3	[C#] protected StateBag ViewState {get;}
4	[C++] protected:property StateBag* get_ViewState();
5	[VB] Protected ReadOnly Property ViewState As StateBag
6	[JScript] protected function get ViewState(): StateBag;
7	
8	Description
9	Gets the statebag for the System.Web.UI.WebControls.DataGridColumn
10	. This property is read-only.
11	Visible
12	TrackViewState
13	
14	[C#] public bool Visible {get; set;}
15	[C++] public:property bool get_Visible();public:property void
16	set_Visible(bool);
17	[VB] Public Property Visible As Boolean
18	[JScript] public function get Visible(): Boolean; public function set
19	Visible(Boolean);
20	
21	Description
22	Gets or sets a value that indicates whether the column is visible in the
23	System.Web.UI.WebControls.DataGrid control.
24	
25	

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Use the System. Web. UI. Web Controls. Data Grid Column. Visible property to programmatically control whether the column is visible in the System.Web.UI.WebControls.DataGrid control. Initialize [C#] public virtual void Initialize(); [C++] public: virtual void Initialize(); [VB] Overridable Public Sub Initialize() [JScript] public function Initialize(); Description **InitializeCell** [C#] public virtual void InitializeCell(TableCell cell, int columnIndex, ListItemType itemType); [C++] public: virtual void InitializeCell(TableCell\* cell, int columnIndex, ListItemType itemType); [VB] Overridable Public Sub InitializeCell(ByVal cell As TableCell, ByVal columnIndex As Integer, ByVal itemType As ListItemType) [JScript] public function InitializeCell(cell: TableCell, columnIndex: int, itemType : ListItemType); Description Initializes a cell in the System. Web. UI. Web Controls. Data Grid Column .

A System.Web.UI.WebControls.TableCell that contains information about the

cell. The column number of the cell to initialize. One of the System.Web.UI.WebControls.ListItemType values. 2 LoadViewState 3 [C#] protected virtual void LoadViewState(object savedState); 5 [C++] protected: virtual void LoadViewState(Object\* savedState); [VB] Overridable Protected Sub LoadViewState(ByVal savedState As Object) [JScript] protected function LoadViewState(savedState : Object); 8 9 Description 10 Loads the state of the System. Web. UI. WebControls. Data Grid Column . 11 An System. Object that contains the saved state of the 12 System. Web. UI. Web Controls. Data Grid Column.13 OnColumnChanged 14 15 [C#] protected virtual void OnColumnChanged(); 16 [C++] protected: virtual void OnColumnChanged(); 17 [VB] Overridable Protected Sub OnColumnChanged() 18 [JScript] protected function OnColumnChanged(); 19 20 Description 21 Raises the ColumnChanged event of a 22 System.Web.UI.WebControls.DataGridColumn object. 23 SaveViewState 24 25

1	
2	[C#] protected virtual object SaveViewState();
3	[C++] protected: virtual Object* SaveViewState();
4	[VB] Overridable Protected Function SaveViewState() As Object
5	[JScript] protected function SaveViewState() : Object;
6	
7	Description
8	Saves the current state of the
9	System.Web.UI.WebControls.DataGridColumn .
10	Return Value: An System.Object that contains the saved state of the
11	System.Web.UI.WebControls.DataGridColumn .
12	IStateManager.LoadViewState
13	
14	[C#] void IStateManager.LoadViewState(object state);
15	[C++] void IStateManager::LoadViewState(Object* state);
16	[VB] Sub LoadViewState(ByVal state As Object) Implements
17	IStateManager.LoadViewState
18	[JScript] function IStateManager.LoadViewState(state : Object);
19	IStateManager.SaveViewState
20	
21	[C#] object IStateManager.SaveViewState();
22	[C++] Object* IStateManager::SaveViewState();
23	[VB] Function SaveViewState() As Object Implements
24	IStateManager.SaveViewState
25	[JScript] function IStateManager.SaveViewState() : Object;

1	IStateManager.TrackViewState
2	
3	[C#] void IStateManager.TrackViewState();
4	[C++] void IStateManager::TrackViewState();
5	[VB] Sub TrackViewState() Implements IStateManager.TrackViewState
6	[JScript] function IStateManager.TrackViewState();
7	ToString
8	
9	[C#] public override string ToString();
10	[C++] public: String* ToString();
11	[VB] Overrides Public Function ToString() As String
12	[JScript] public override function ToString() : String;
13	
14	Description
15	Returns the string representation of the column.
16	Return Value: Returns System.String.Empty.
17	Use the System.Web.UI.WebControls.DataGridColumn.ToString
18	method to get the string representation of the column.
19	TrackViewState
20	
21	[C#] protected virtual void TrackViewState();
22	[C++] protected: virtual void TrackViewState();
23	[VB] Overridable Protected Sub TrackViewState()
24	[JScript] protected function TrackViewState();
25	

Description

3

4

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Marks the starting point to begin tracking and saving changes to the control as part of the control viewstate.

DataGridColumnCollection class (System.Web.UI.WebControls)

TrackViewState

Description

A collection of **System.Web.UI.WebControls.DataGridColumn** derived column objects that represent the columns in a

System.Web.UI.WebControls.DataGrid control. This class cannot be inherited.

Use the **System.Web.UI.WebControls.DataGridColumnCollection** to programmatically manage a collection of

System.Web.UI.WebControls.DataGridColumn derived column objects. These objects represent the columns in a System.Web.UI.WebControls.DataGrid control. You can add, remove, or insert columns into the

 $System. Web. UI. WebControls. Data Grid Column Collection \ .$ 

DataGridColumnCollection

Example Syntax:

**TrackViewState** 

[C#] public DataGridColumnCollection(DataGrid owner, ArrayList columns); [C++] public: DataGridColumnCollection(DataGrid\* owner, ArrayList\*

columns);

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

[VB] Public Sub New(ByVal owner As DataGrid, ByVal columns As ArrayList) [JScript] public function DataGridColumnCollection(owner: DataGrid, columns: ArrayList); Description Initializes a new instance of the  $System. Web. UI. Web Controls. Data Grid Column Collection \ class.$ Use this constructor to create and initialize a new instance of the System. Web. UI. Web Controls. Data Grid Column Collection class. The System.Web.UI.WebControls.DataGrid control that corresponds with this collection. A System. Collections. ArrayList object that stores the collection of columns. Count **TrackViewState** [C#] public int Count {get;} [C++] public: \_\_property int get\_Count(); [VB] Public ReadOnly Property Count As Integer [JScript] public function get Count(): int; Description Gets the number of columns in the  $System. Web. UI. Web Controls. Data Grid Column Collection \ . \\$ Use this property to determine the number of columns in the

 $System. Web. UI. Web Controls. Data Grid Column Collection \ . \ The$ 

1	System.Web.UI.WebControls.DataGridColumnCollection.Count property is
2	commonly used when iterating through the collection to determine the upper
3	bound of the collection.
4	IsReadOnly
5	TrackViewState
6	
7	[C#] public bool IsReadOnly {get;}
8	[C++] public:property bool get_IsReadOnly();
9	[VB] Public ReadOnly Property IsReadOnly As Boolean
10	[JScript] public function get IsReadOnly(): Boolean;
11	
12	Description
13	Gets a value that indicates whether the columns in the
14	System.Web.UI.WebControls.DataGridColumnCollection can be modified.
15	This property always returns false to indicate that the
16	System.Web.UI.WebControls.DataGridColumnCollection can be written to in
17	all cases.
18	IsSynchronized
19	TrackViewState
20	
21	[C#] public bool IsSynchronized {get;}
22	[C++] public:property bool get_IsSynchronized();
23	[VB] Public ReadOnly Property IsSynchronized As Boolean
24	[JScript] public function get IsSynchronized(): Boolean;
25	

4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13

1

2

3

15

14

17 18

19 20

21

23

24

25

Description

Gets a value indicating whether access to the System.Web.UI.WebControls.DataGridColumnCollection is synchronized (thread-safe).

This property is derived from the **System.Collections.ICollection** class and is overridden to always return **false**.

Item

TrackViewState

[C#] public DataGridColumn this[int index] {get;}

[C++] public: \_\_property DataGridColumn\* get\_Item(int index);

[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As

DataGridColumn

 $[JScript]\ return Value = Data Grid Column Collection Object. Item (index);$ 

Description

Gets a System.Web.UI.WebControls.DataGridColumn derived column object from the System.Web.UI.WebControls.DataGridColumnCollection collection at the specified index.

Use this indexer to get a **System.Web.UI.WebControls.DataGridColumn** derived column object from the

System.Web.UI.WebControls.DataGridColumnCollection at the specified index using array notation. The index of the

System.Web.UI.WebControls.DataGridColumn derived object in the

1	System.Web.UI.WebControls.DataGridColumnCollection collection to
2	retrieve.
3	SyncRoot
4	TrackViewState
5	
6	[C#] public object SyncRoot {get;}
7	[C++] public:property Object* get_SyncRoot();
8	[VB] Public ReadOnly Property SyncRoot As Object
9	[JScript] public function get SyncRoot() : Object;
10	
11	Description
12	Gets the object that can be used to synchronize access to the
13	System.Web.UI.WebControls.DataGridColumnCollection .
14	The object returned in this implementation is the
15	System.Web.UI.WebControls.DataGridColumnCollection object.
16	Add
17	
18	[C#] public void Add(DataGridColumn column);
19	[C++] public: void Add(DataGridColumn* column);
20	[VB] Public Sub Add(ByVal column As DataGridColumn)
21	[JScript] public function Add(column : DataGridColumn);
22	
23	Description
24	
25	

Appends the specified **System.Web.UI.WebControls.DataGridColumn** derived column object to the end of the

 $System. Web. UI. Web Controls. Data Grid Column Collection \ .$ 

Use this method to add a

System.Web.UI.WebControls.DataGridColumn derived column object to the end of a System.Web.UI.WebControls.DataGridColumnCollection . The System.Web.UI.WebControls.DataGridColumn derived column object to append to the System.Web.UI.WebControls.DataGridColumnCollection.

AddAt

[C#] public void AddAt(int index, DataGridColumn column);

[C++] public: void AddAt(int index, DataGridColumn\* column);

[VB] Public Sub AddAt(ByVal index As Integer, ByVal column As

DataGridColumn)

[JScript] public function AddAt(index: int, column: DataGridColumn);

Description

Inserts a System.Web.UI.WebControls.DataGridColumn derived column object in the System.Web.UI.WebControls.DataGridColumnCollection at the specified index.

Use this method to insert a

System.Web.UI.WebControls.DataGridColumn derived column object at the specified index location in the

 $System. Web. UI. Web Controls. Data Grid Column Collection \ . \ The \ index \ location$  in the System. Web. UI. Web Controls. Data Grid Column Collection to insert the

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

System. Web. UI. Web Controls. Data Grid Column derived column object. The System. Web. UI. Web Controls. Data Grid Column derived column object to insert  $into\ the\ System. Web. UI. Web Controls. Data Grid Column Collection.$ Clear [C#] public void Clear(); [C++] public: void Clear(); [VB] Public Sub Clear() [JScript] public function Clear(); Description Removes all System. Web. UI. WebControls. Data Grid Column derived column objects from the  $System. Web. UI. Web Controls. Data Grid Column Collection \ .$ Use this method to remove all System.Web.UI.WebControls.DataGridColumn derived column objects from  $the \ System. Web. UI. Web Controls. Data Grid Column Collection \ .$ CopyTo [C#] public void CopyTo(Array array, int index); [C++] public: sealed void CopyTo(Array\* array, int index); [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As Integer) [JScript] public function CopyTo(array: Array, index: int);

#### Description

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Copies the items from the

System.Web.UI.WebControls.DataGridColumnCollection to the specified System.Array object, starting at the specified index in the System.Array object.

Use this method to copy the contents of the

System.Web.UI.WebControls.DataGridColumnCollection into the specified System.Array object, starting at the specified index. A zero-based System.Array object that receives the copied items from the

System.Web.UI.WebControls.DataGridColumnCollection. The first position in the specified System.Array object to receive the copied contents.

GetEnumerator

[C#] public IEnumerator GetEnumerator();

[C++] public: sealed IEnumerator\* GetEnumerator();

[VB] NotOverridable Public Function GetEnumerator() As IEnumerator

[JScript] public function GetEnumerator(): IEnumerator;

### Description

Returns an System.Collections.IEnumerator interface that contains all System.Web.UI.WebControls.DataGridColumn derived column objects in the System.Web.UI.WebControls.DataGridColumnCollection .

Return Value: A System.Collections.IEnumerator interface that contains all System.Web.UI.WebControls.DataGridColumn derived column objects in the

System. Web. UI. Web Controls. Data Grid Column Collection.

lee@hayes pilc 509-324-9256 1743 MS1-863US.APP

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use this method to create a **System.Collections.IEnumerator** that can be iterated through easily to get each item in the

 $System. Web. UI. Web Controls. Data Grid Column Collection \ .$ 

IndexOf

[C#] public int IndexOf(DataGridColumn column);

[C++] public: int IndexOf(DataGridColumn\* column);

[VB] Public Function IndexOf(ByVal column As DataGridColumn) As Integer

[JScript] public function IndexOf(column : DataGridColumn) : int;

Description

Returns the index of the specified

 ${\bf System. Web. UI. WebControls. Data Grid Column\ derived\ column\ object\ from\ the\ System. Web. UI. WebControls. Data Grid Column Collection\ .}$ 

Return Value: The index position of the specified

System.Web.UI.WebControls.DataGridColumn derived column object in the System.Web.UI.WebControls.DataGridColumnCollection . The default value is -1 , which indicates that the specified

System.Web.UI.WebControls.DataGridColumn derived object is not found.

Use this method to determine the index number of the specified System.Web.UI.WebControls.DataGridColumn derived column object in the System.Web.UI.WebControls.DataGridColumnCollection . If the specified System.Web.UI.WebControls.DataGridColumn derived column object is not found, an index of -1 is returned. The

lee@haves ₀lic 509-324-9256 1744

1	System.Web.UI.WebControls.DataGridColumn derived column object to
2	search for in the System. Web. UI. WebControls. Data Grid Column Collection.
3	Remove
4	
5	[C#] public void Remove(DataGridColumn column);
6	[C++] public: void Remove(DataGridColumn* column);
7	[VB] Public Sub Remove(ByVal column As DataGridColumn)
8	[JScript] public function Remove(column : DataGridColumn);
9	
10	Description
11	Removes the specified System.Web.UI.WebControls.DataGridColumn
12	derived column object from the
13	System.Web.UI.WebControls.DataGridColumnCollection .
14	Use this method to remove the specified
15	System.Web.UI.WebControls.DataGridColumn derived column object from a
16	System.Web.UI.WebControls.DataGridColumnCollection . The
17	System.Web.UI.WebControls.DataGridColumn derived column object to
18	remove from the System.Web.UI.WebControls.DataGridColumnCollection.
19	RemoveAt
20	
21	[C#] public void RemoveAt(int index);
22	[C++] public: void RemoveAt(int index);
23	[VB] Public Sub RemoveAt(ByVal index As Integer)
24	[JScript] public function RemoveAt(index : int);
25	

ا '	֡
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	

20

21

22

23

24

25

_	_	_
Des	ovin	tion
Des	cr ip	$\iota\iota\iota O \iota\iota$

1

2

Removes a **System.Web.UI.WebControls.DataGridColumn** derived column object from the

System.Web.UI.WebControls.DataGridColumnCollection at the specified index.

Use this method to remove a

System.Web.UI.WebControls.DataGridColumn derived column object from a System.Web.UI.WebControls.DataGridColumnCollection at the specified index. The index of the System.Web.UI.WebControls.DataGridColumn derived column object in the

 ${\bf System. Web. UI. WebControls. Data Grid Column Collection}\ to\ remove.$ 

IStateManager.LoadViewState

[C#] void IStateManager.LoadViewState(object savedState);

[C++] void IStateManager::LoadViewState(Object\* savedState);

[VB] Sub LoadViewState(ByVal savedState As Object) Implements

IS tate Manager. Load View State

 $[JScript]\ function\ IState Manager. Load View State (saved State: Object);$ 

IS tate Manager. Save View State

[C#] object IStateManager.SaveViewState();

[C++] Object\* IStateManager::SaveViewState();

[VB] Function SaveViewState() As Object Implements

IStateManager.SaveViewState
[JScript] function IStateManager.SaveViewState(): Object;
IStateManager.TrackViewState
[C#] void IStateManager.TrackViewState();
[C++] void IStateManager::TrackViewState();
[VB] Sub TrackViewState() Implements IStateManager.TrackViewState
[JScript] function IStateManager.TrackViewState();
DataGridCommandEventArgs class (System.Web.UI.WebControls)
ToString
Description
Provides data for the
System.Web.UI.WebControls.DataGrid.CancelCommand,
System.Web.UI.WebControls.DataGrid.DeleteCommand,
System.Web.UI.WebControls.DataGrid.EditCommand,
System.Web.UI.WebControls.DataGrid.ItemCommand, and
System.Web.UI.WebControls.DataGrid.UpdateCommand events of the
System.Web.UI.WebControls.DataGrid control. This class cannot be inherited.
The System.Web.UI.WebControls.DataGrid.CancelCommand event is
raised when the Cancel button for an item in the
System.Web.UI.WebControls.DataGrid control is clicked.
DataGridCommandEventArgs
Example Syntax:

## **ToString**

2

1

3

5

7

8

9

11

10

12 13

14

16

15

17 18

19

20

21 22

23

24

25

[C#] public DataGridCommandEventArgs(DataGridItem item, object commandSource, CommandEventArgs originalArgs);

[C++] public: DataGridCommandEventArgs(DataGridItem\* item, Object\* commandSource, CommandEventArgs\* originalArgs);

[VB] Public Sub New(ByVal item As DataGridItem, ByVal commandSource As Object, ByVal originalArgs As CommandEventArgs)

[JScript] public function DataGridCommandEventArgs(item: DataGridItem, commandSource : Object, originalArgs : CommandEventArgs);

Description

Initializes a new instance of the

 $System. Web. UI. Web Controls. Data Grid Command Event Args\ class.$ 

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.DataGridCommandEventArgs class. A System.Web.UI.WebControls.DataGridItem that represents the selected item in the System. Web. UI. Web Controls. Data Grid. The source of the command. A System.Web.UI.WebControls.CommandEventArgs that contains the event data.

CommandArgument

CommandName

CommandSource

**ToString** 

3	Description
4	Gets the sour
5	Use the
6	System.Web.UI.W
7	ce property to deter
8	is commonly used t
9	Item
10	ToString
11	
12	[C#] public DataGr
13	[C++] public:pro
14	[VB] Public ReadC
15	[JScript] public fun
16	
17	Description
18	Gets the iter
19	System.Web.UI.W
20	The System
21	property is used to
22	System.Web.UI.W
23	DataGridCo

rce of the command.

ebControls.DataGridCommandEventArgs.CommandSour mine the command source that raised the event. This property o determine which command raised the event.

ridItem Item {get;}

operty DataGridItem\* get\_Item();

Only Property Item As DataGridItem

nction get Item(): DataGridItem;

m containing the command source in the

VebControls.DataGrid control.

. We b. UI. We bC on trols. Data Grid Command Event Args. Itemaccess the properties of the selected item in the

VebControls.DataGrid control.

ommandEventHandler delegate (System.Web.UI.WebControls)

**ToString** 

1749

MS1-863US.APP

Description

event data.

Represents the method that will handle the

System.Web.UI.WebControls.DataGrid.CancelCommand,

System.Web.UI.WebControls.DataGrid.DeleteCommand,

System.Web.UI.WebControls.DataGrid.EditCommand,

System.Web.UI.WebControls.DataGrid.ItemCommand, and

System.Web.UI.WebControls.DataGrid.UpdateCommand events of a

System.Web.UI.WebControls.DataGrid. The source of the event. A

System.Web.UI.WebControls.DataGridCommandEventArgs that contains the

When you create a

System.Web.UI.WebControls.DataGridCommandEventHandler delegate, you identify the method that will handle the event. To associate the event with your event handler, add an instance of the delegate to the event. The event handler is called whenever the event occurs, unless you remove the delegate. For more information about event handler delegates, see .

DataGridItem class (System.Web.UI.WebControls)
ToString

Description

Represents an item (row) in the **System.Web.UI.WebControls.DataGrid** control.

A System.Web.UI.WebControls.DataGridItem object represents an item (row) in the System.Web.UI.WebControls.DataGrid control, such as the heading section, the footer section, or a data row.

DataGridItem

Example Syntax:

ToString

[C#] public DataGridItem(int itemIndex, int dataSetIndex, ListItemType itemType);

[C++] public: DataGridItem(int itemIndex, int dataSetIndex, ListItemType itemType);

[VB] Public Sub New(ByVal itemIndex As Integer, ByVal dataSetIndex As Integer, ByVal itemType As ListItemType)

[JScript] public function DataGridItem(itemIndex : int, dataSetIndex : int, itemType : ListItemType);

## Description

Initializes a new instance of the

 ${\bf System. Web. UI. Web Controls. Data Grid Item\ class.}$ 

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.DataGridItem class. The index of the item in the System.Web.UI.WebControls.DataGrid control from the System.Web.UI.WebControls.DataGrid.Items collection. The index number of the item, from the bound data source, that appears in the

lee@hayes pilc 509-324-9256 1751 MS1-863US.APP

3

4

5

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

System.Web.UI.WebControls.DataGrid control. One of the  $System. Web. UI. Web Controls. List I tem Type \ values.$ AccessKey Attributes BackColor BorderColor BorderStyle **BorderWidth** Cells ChildControlsCreated ClientID Context Controls ControlStyle ControlStyleCreated CssClass DataItem **ToString** Description Gets or sets the data item represented by the System.Web.UI.WebControls.DataGridItem object in the

System.Web.UI.WebControls.DataGrid control.

Use the System.Web.UI.WebControls.DataGridItem.DataItem property
to specify or determine the properties of a data item represented by the
System.Web.UI.WebControls.DataGridItem object in the
System.Web.UI.WebControls.DataGrid control.
DataSetIndex
ToString
[C#] public virtual int DataSetIndex {get;}
[C++] public:property virtual int get_DataSetIndex();
[VB] Overridable Public ReadOnly Property DataSetIndex As Integer
[JScript] public function get DataSetIndex(): int;
Description
Gets the index number the System.Web.UI.WebControls.DataGridItem
object from the bound data source.
Use the System.Web.UI.WebControls.DataGridItem.DataSetIndex
property to the index number of the
System.Web.UI.WebControls.DataGridItem object from the bound data source
Enabled
EnableViewState
Events
Font
ForeColor
HasChildViewState
Height

HorizontalAlign ID IsTrackingViewState 3 ItemIndex 4 **ToString** 5 6 7 Description 8 Gets the index of the System. Web.UI. WebControls. Data Grid I tem object 9 from the System. Web. UI. Web Controls. Data Grid. Items collection of the 10 System.Web.UI.WebControls.DataGrid control. 11  $Use\ the\ System. Web. UI. WebControls. Data Grid Item. Item Index$ 12 property to determine the index number of the 13 System.Web.UI.WebControls.DataGridItem object from the 14 System.Web.UI.WebControls.DataGrid.Items collection of the 15 System.Web.UI.WebControls.DataGrid control. 16 ItemType 17 **ToString** 18 19 [C#] public virtual ListItemType ItemType {get;} 20 [C++] public: property virtual ListItemType get\_ItemType(); 21 [VB] Overridable Public ReadOnly Property ItemType As ListItemType 22 [JScript] public function get ItemType(): ListItemType; 23 24 Description

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets the type of the item represented by the System.Web.UI.WebControls.DataGridItem object in the System.Web.UI.WebControls.DataGrid control.

Use the **System.Web.UI.WebControls.DataGridItem.ItemType** property to determine the type of an item in the **System.Web.UI.WebControls.DataGrid** control. The following table lists the various item types.

NamingContainer

Page

Parent

Site

Style

TabIndex

TagKey

TagName

**TemplateSourceDirectory** 

ToolTip

UniqueID

VerticalAlign

ViewState

ViewStateIgnoresCase

Visible

Width

OnBubbleEvent

[C#] protected override bool OnBubbleEvent(object source, EventArgs e);

1	[C++] protected: bool OnBubbleEvent(Object* source, EventArgs* e);
2	[VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,
3	ByVal e As EventArgs) As Boolean
4	[JScript] protected override function OnBubbleEvent(source : Object, e :
5	EventArgs): Boolean;
6	
7	Description
8	SetItemType
9	
10	[C#] protected internal virtual void SetItemType(ListItemType itemType);
11	[C++] protected public: virtual void SetItemType(ListItemType itemType);
12	[VB] Overridable Protected Friend Dim Sub SetItemType(ByVal itemType As
13	ListItemType)
14	[JScript] package function SetItemType(itemType : ListItemType);
15	
16	Description
17	DataGridItemCollection class (System.Web.UI.WebControls)
18	TrackViewState
19	
20	
21	Description
22	Represents a collection of System.Web.UI.WebControls.DataGridItem
23	objects in a System.Web.UI.WebControls.DataGrid control.
24	The System.Web.UI.WebControls.DataGridItemCollection class
25	represents a collection of System.Web.UI.WebControls.DataGridItem objects

1	which in turn represent the data items in a
2	System.Web.UI.WebControls.DataGrid control. To programmatically retrieve
3	System.Web.UI.WebControls.DataGridItem objects from a
4	System.Web.UI.WebControls.DataGrid control, use one of following methods:
5	Use the indexer to get a single System.Web.UI.WebControls.DataGridItem
6	object from the collection, using array notation.
7	DataGridItemCollection
8	Example Syntax:
9	TrackViewState
10	
11	[C#] public DataGridItemCollection(ArrayList items);
12	[C++] public: DataGridItemCollection(ArrayList* items);
13	[VB] Public Sub New(ByVal items As ArrayList)
14	[JScript] public function DataGridItemCollection(items : ArrayList);
15	
16	Description
17	Initializes a new instance of the
18	System.Web.UI.WebControls.DataGridItemCollection class.
19	Use this constructor to create and initialize a new instance of the
20	System.Web.UI.WebControls.DataGridItemCollection class. A
21	System.Collections.ArrayList object that contains the items with which to
22	initialize the collection.
23	Count
24	TrackViewState
25	

1 [C#] public int Count {get;} [C++] public: property int get\_Count(); 3 [VB] Public ReadOnly Property Count As Integer [JScript] public function get Count(): int; 5 6 Description 7 Gets the number of System. Web. UI. Web Controls. Data Grid I tem objects 8 in the collection. 9 Use the System. Web. UI. Web Controls. Data Grid I tem Collection. Count 10 property to determine the number of 11 System.Web.UI.WebControls.DataGridItem objects in the 12  $System. Web. UI. Web Controls. Data Grid Item Collection \ collection. \ The$ 13  ${\bf System. Web. UI. WebControls. Data Grid Item Collection. Count\ property\ is}$ 14 commonly used when iterating through the collection to determine the upper 15 bound of the collection. 16 IsReadOnly 17 **TrackViewState** 18 19 [C#] public bool IsReadOnly {get;} 20 [C++] public: \_\_property bool get\_IsReadOnly(); 21 [VB] Public ReadOnly Property IsReadOnly As Boolean 22 [JScript] public function get IsReadOnly(): Boolean; 23 24 Description

1	Gets a value that indicates whether the
2	System.Web.UI.WebControls.DataGridItem objects in the
3	System.Web.UI.WebControls.DataGridItemCollection can be modified.
4	This property always returns false to indicate that the
5	System.Web.UI.WebControls.DataGridItemCollection can be written to in all
6	cases.
7	IsSynchronized
8	TrackViewState
9	
10	[C#] public bool IsSynchronized {get;}
11	[C++] public:property bool get_IsSynchronized();
12	[VB] Public ReadOnly Property IsSynchronized As Boolean
13	[JScript] public function get IsSynchronized(): Boolean;
14	
15	Description
16	Gets a value indicating whether access to the
17	System.Web.UI.WebControls.DataGridItemCollection is synchronized
18	(thread-safe).
19	This property is derived from the System.Collections.ICollection class and
20	is overridden to always return false.
21	Item
22	TrackViewState
23	
24	[C#] public DataGridItem this[int index] {get;}
25	[C++] public:property DataGridItem* get_Item(int index);

[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As DataGridItem [JScript] returnValue = DataGridItemCollectionObject.Item(index); 3 4 Description 5 Gets the System. Web.UI. WebControls. Data Grid Item object at the 6 specified index in the collection. 7 Use this indexer to get a System. Web.UI. WebControls. Data Grid I tem 8 object from the System. Web. UI. WebControls. Data Grid I temCollection at the 9 specified index, using array notation. The zero-based index of the 10 System. Web. UI. Web Controls. Data Grid I tem object to retrieve from the 11 collection. 12 SyncRoot 13 **TrackViewState** 14 15 [C#] public object SyncRoot {get;} 16 [C++] public: \_\_property Object\* get\_SyncRoot(); 17 [VB] Public ReadOnly Property SyncRoot As Object 18 [JScript] public function get SyncRoot(): Object; 19 20 Description 21 Gets the object that can be used to synchronize access to the 22  $System. Web. UI. Web Controls. Data Grid Item Collection \ .$ 23 The object returned in this implementation is the 24  ${\bf System. Web. UI. WebControls. Data Grid Item Collection}\ object\ itself.$ 

1	СоруТо
2	
3	[C#] public void CopyTo(Array array, int index);
4	[C++] public:sealed void CopyTo(Array* array, int index);
5	[VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
6	Integer)
7	[JScript] public function CopyTo(array: Array, index: int);
8	
9	Description
10	Copies all the items from this
11	System.Web.UI.WebControls.DataGridItemCollection to the specified
12	System.Array object, starting at the specified index in the System.Array object.
13	Use this method to copy the contents of the
14	System.Web.UI.WebControls.DataGridItemCollection collection into the
15	specified System.Array object, starting at the specified index. A zero-based
16	System.Array object that receives the copied items from the
17	System.Web.UI.WebControls.DataGridItemCollection. The first position in the
18	specified System.Array object to receive the copied contents.
19	GetEnumerator
20	
21	[C#] public IEnumerator GetEnumerator();
22	[C++] public:sealed IEnumerator* GetEnumerator();
23	[VB] NotOverridable Public Function GetEnumerator() As IEnumerator
24	[JScript] public function GetEnumerator(): IEnumerator;

$\sim$	escription	
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	acammet ar	1
,,	excriminor	ı

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Returns a System.Collections.IEnumerator interface that contains all System.Web.UI.WebControls.DataGridItem objects in the System.Web.UI.WebControls.DataGridItemCollection .

Return Value: A System.Collections.IEnumerator interface that contains all System.Web.UI.WebControls.DataGridItem objects in the

Use this method to create a **System.Collections.IEnumerator** that can be easily iterated through to get each item in the

 $System. Web. UI. Web Controls. Data Grid Item Collection \ .$ 

System.Web.UI.WebControls.DataGridItemCollection .

DataGridItemEventArgs class (System.Web.UI.WebControls)
ToString

## Description

Provides data for the

System.Web.UI.WebControls.DataGrid.ItemCreated and

System.Web.UI.WebControls.DataGrid.ItemDataBound events of the

System.Web.UI.WebControls.DataGrid control. This class cannot be inherited.

The System.Web.UI.WebControls.DataGrid.ItemCreated event is raised when an item in the System.Web.UI.WebControls.DataGrid control is created.

DataGridItemEventArgs

Example Syntax:

1	ToString
2	
3	[C#] public DataGridItemEventArgs(DataGridItem item);
4	[C++] public: DataGridItemEventArgs(DataGridItem* item);
5	[VB] Public Sub New(ByVal item As DataGridItem)
6	[JScript] public function DataGridItemEventArgs(item: DataGridItem);
7	
8	Description
9	Initializes a new instance of
10	System.Web.UI.WebControls.DataGridItemEventArgs class.
11	Use this constructor to create and initialize a new instance of the
12	System.Web.UI.WebControls.DataGridItemEventArgs class. A
13	System.Web.UI.WebControls.DataGridItem that represents an item in the
14	System.Web.UI.WebControls.DataGrid.
15	Item
16	ToString
17	
18	[C#] public DataGridItem Item {get;}
19	[C++] public:property DataGridItem* get_Item();
20	[VB] Public ReadOnly Property Item As DataGridItem
21	[JScript] public function get Item(): DataGridItem;
22	
23	Description
24	Gets the referenced item in the System.Web.UI.WebControls.DataGrid
25	control when the event is raised.

data.

Use this property to programmatically access the item referenced in the System.Web.UI.WebControls.DataGrid control when the event is raised.

DataGridItemEventHandler delegate (System.Web.UI.WebControls)

ToString

Description

Represents the method that will handle the

System.Web.UI.WebControls.DataGrid.ItemCreated and

System.Web.UI.WebControls.DataGrid.ItemDataBound events of a

System.Web.UI.WebControls.DataGrid . The source of the event. A

The System.Web.UI.WebControls.DataGrid.ItemCreated event is raised when an item in the System.Web.UI.WebControls.DataGrid control is created.

 $System. Web. UI. Web Controls. Data Grid I tem Event Args\ than\ contains\ the\ event are contains\ the\ event are contains. The properties of the control of the control of the contro$ 

DataGridPageChangedEventArgs class (System.Web.UI.WebControls)
ToString

Description

Provides data for the

System.Web.UI.WebControls.DataGrid.PageIndexChanged event of the System.Web.UI.WebControls.DataGrid control. This class cannot be inherited.

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

The System.Web.UI.WebControls.DataGrid.PageIndexChanged event is raised when a button in the page selection element of the System.Web.UI.WebControls.DataGrid control is clicked. DataGridPageChangedEventArgs Example Syntax: **ToString** [C#] public DataGridPageChangedEventArgs(object commandSource, int newPageIndex); [C++] public: DataGridPageChangedEventArgs(Object\* commandSource, int newPageIndex); [VB] Public Sub New(ByVal commandSource As Object, ByVal newPageIndex As Integer)  $[JScript]\ public\ function\ Data GridPage Change dEventArgs (command Source: Stript)]\ and the properties of the prop$ Object, newPageIndex: int); Description Initializes a new instance of the  $System. Web. UI. WebControls. Data Grid Page Change d Event Args\ class.$ Use this constructor to create and initialize a new instance of the  $System. Web. UI. WebControls. Data Grid Page Change d Event Args\ class.\ The$ source of the command. The index of the page selected by the user from the page selection element of the System. Web. UI. WebControls. Data Grid control.

CommandSource

**ToString** 

```
1
            [C#] public object CommandSource {get;}
            [C++] public: __property Object* get_CommandSource();
3
            [VB] Public ReadOnly Property CommandSource As Object
            [JScript] public function get CommandSource(): Object;
 5
 6
            Description
 7
                                 Gets the source of the command.
 8
                                 Use the
 9
             System. Web. UI. Web Controls. Data Grid Page Changed Event Args. Command States and States are also as a first of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Control of the Contr
10
             ource property to determine the source of the command that raised the event.
11
             Because this event is only raised when a button in the page selection element of
12
             the System. Web.UI. WebControls. Data Grid control is clicked, the returned
13
             source is a System. Web.UI. WebControls. Data Grid I tem that represents the page
14
              selection element.
15
                                   NewPageIndex
 16
                                   ToString
 17
 18
              [C#] public int NewPageIndex {get;}
 19
              [C++] public: __property int get_NewPageIndex();
 20
              [VB] Public ReadOnly Property NewPageIndex As Integer
 21
               [JScript] public function get NewPageIndex(): int;
 22
  23
               Description
  24
  25
```

24

25

2

3

5

8

9

Gets the index of the page selected by the user in the page selection element of the System. Web. UI. Web Controls. Data Grid control.

Use the

System. Web. UI. WebControls. Data Grid Page Change d Event Args. New Page Index and the property of the proex property to determine the index of the page selected by the user in the page selection element of the System. Web. UI. Web Controls. Data Grid control. This value is often used to set the

System.Web.UI.WebControls.DataGrid.CurrentPageIndex property of the System. Web. UI. Web Controls. Data Grid control to display the selected page.

DataGridPageChangedEventHandler delegate (System.Web.UI.WebControls)

**ToString** 

Description

Represents the method that will handle the

System.Web.UI.WebControls.DataGrid.PageIndexChanged event of the System.Web.UI.WebControls.DataGrid control. The source of the event. A  $System. Web. UI. WebControls. Data Grid Page Changed Event Args\ that\ contains$ the event data.

The System.Web.UI.WebControls.DataGrid.PageIndexChanged event is raised when a button in the page selection element of the

System.Web.UI.WebControls.DataGrid control is clicked.

DataGridPagerStyle class (System.Web.UI.WebControls)

**ToString** 

Description
Spec

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Specifies the style for the pager of the

System. Web.UI. WebControls. DataGrid control. This class cannot be inherited.

The pager is an element on the **System.Web.UI.WebControls.DataGrid** control that allow you to link to other pages when paging is enabled. The **System.Web.UI.WebControls.DataGrid.PagerStyle** property of the **System.Web.UI.WebControls.DataGrid** uses an instance of this class to represent the style properties for the pager.

BackColor

BorderColor

BorderStyle

BorderWidth

Container

CssClass

DesignMode

**Events** 

Font

ForeColor

Height

HorizontalAlign

**IsEmpty** 

**IsTrackingViewState** 

Mode

**ToString** 

Description

Gets or sets a value that specifies whether the pager element displays buttons that link to the next and previous page, or numeric buttons that link directly to a page.

Use the **System.Web.UI.WebControls.DataGridPagerStyle.Mode** property to specify which set of pager buttons to use on the

System.Web.UI.WebControls.DataGrid control. You can specify buttons that link to the next and previous page, or numeric buttons that link directly to a page.

NextPageText

**ToString** 

[C#] public string NextPageText {get; set;}

[C++] public: \_\_property String\* get\_NextPageText();public: \_\_property void
set NextPageText(String\*);

[VB] Public Property NextPageText As String

[JScript] public function get NextPageText() : String; public function set NextPageText(String);

Description

21

22

23

24

Gets or sets the text displayed for the next page button.

Use the

System.Web.UI.WebControls.DataGridPagerStyle.NextPageText property to

1	provide custom text for the next page button. The
2	System.Web.UI.WebControls.DataGridPagerStyle.Mode property must be set
3	to PagerMode.NextPrev for this property to have any effect.
4	PageButtonCount
5	ToString
6	
7	[C#] public int PageButtonCount {get; set;}
8	[C++] public:property int get_PageButtonCount();public:property void
9	set_PageButtonCount(int);
10	[VB] Public Property PageButtonCount As Integer
11	[JScript] public function get PageButtonCount(): int;public function set
12	PageButtonCount(int);
13	
14	Description
15	Gets or sets the number of numeric buttons to display concurrently in the
16	pager element of the System.Web.UI.WebControls.DataGrid control.
17	Use the
18	System.Web.UI.WebControls.DataGridPagerStyle.PageButtonCount property
19	to specify the number of numeric buttons to display concurrently in the pager
20	element of the System.Web.UI.WebControls.DataGrid control. The
21	System.Web.UI.WebControls.DataGridPagerStyle.Mode property must be set
22	to PagerMode.NumericPages for this property to have any effect.
23	Position
24	ToString
25	

PrevPageText(String);

```
1
    [C#] public PagerPosition Position {get; set;}
    [C++] public: property PagerPosition get Position();public: property void
    set Position(PagerPosition);
    [VB] Public Property Position As PagerPosition
    [JScript] public function get Position(): PagerPosition; public function set
    Position(PagerPosition);
8
    Description
           Gets or sets the position of the pager element in the
10
    System.Web.UI.WebControls.DataGrid control.
11
           Use the System. Web. UI. Web Controls. Data Grid Pager Style. Position
12
    property to specify the location where the pager element is displayed in the
13
    System.Web.UI.WebControls.DataGrid control. The pager element can be
14
    displayed at the upper edge, the lower edge, or at both the upper and lower edges
15
    of the System.Web.UI.WebControls.DataGrid control.
16
           PrevPageText
17
           ToString
18
19
    [C#] public string PrevPageText {get; set;}
20
    [C++] public: __property String* get_PrevPageText();public: __property void
21
    set PrevPageText(String*);
22
    [VB] Public Property PrevPageText As String
    [JScript] public function get PrevPageText(): String; public function set
```

Description

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the text displayed for the previous page button.

Use the

 ${\bf System. Web. UI. WebControls. Data Grid Pager Style. Prev Page Text\ property\ to}$ provide custom text for the next page button. The

System.Web.UI.WebControls.DataGridPagerStyle.Mode property must be set to PagerMode.NextPrev for this property to have any effect.

Site

VerticalAlign

ViewState

Visible

**ToString** 

Description

Gets or sets a value indicating whether the pager is displayed in the  $System. Web. UI. Web Controls. Data Grid\ control.$ 

 $Use\ the\ System. Web. UI. Web Controls. Data Grid Pager Style. Visible$ property to specify whether the pager is displayed in the

System.Web.UI.WebControls.DataGrid control.

Width

Wrap

CopyFrom

1	
2	[C#] public override void CopyFrom(Style s);
3	[C++] public: void CopyFrom(Style* s);
4	[VB] Overrides Public Sub CopyFrom(ByVal s As Style)
5	[JScript] public override function CopyFrom(s : Style);
6	
7	Description
8	Copies the style of the specified System.Web.UI.WebControls.Style
9	object into this instance of the
10	System.Web.UI.WebControls.DataGridPagerStyle class.
11	Use the
12	System.Web.UI.WebControls.DataGridPagerStyle.CopyFrom(System.Web.U
13	I.WebControls.Style) method to copy the style properties of the specified
14	System.Web.UI.WebControls.Style object into this instance of the
15	System.Web.UI.WebControls.DataGridPagerStyle class. The
16	System.Web.UI.WebControls.Style object to copy from.
17	MergeWith
18	
19	[C#] public override void MergeWith(Style s);
20	[C++] public: void MergeWith(Style* s);
21	[VB] Overrides Public Sub MergeWith(ByVal s As Style)
22	[JScript] public override function MergeWith(s : Style);
23	
24	Description
25	

object with this instance of the System.Web.UI.WebControls.DataGridPagerStyle class. Use the System.Web.UI.WebControls.DataGridPagerStyle.MergeWith(System.Web. UI.WebControls.Style) method to combine the style properties of the specified System. Web. UI. Web Controls. Style object with this instance of the System.Web.UI.WebControls.DataGridPagerStyle class. If a property from this instance is already set to a value, the property is unchanged. If a property is not set, this method sets that property with the value from the corresponding property 10 of the System. Web. UI. Web Controls. Style object. The 11 System.Web.UI.WebControls.Style object to merge with. 12 Reset 13 14 [C#] public override void Reset(); 15 [C++] public: void Reset(); [VB] Overrides Public Sub Reset() 17 [JScript] public override function Reset(); 18 19 Description 20 Restores the System. Web. UI. WebControls. Data Grid Pager Style object 21 to its default values. 22 Use the System. Web. UI. WebControls. Data Grid Pager Style. Reset 23 method to restore the System. Web. UI. Web Controls. Data Grid Pager Style object 24

to its default values.

Merges the style of the specified System.Web.UI.WebControls.Style

1	DataGridSortCommandEventArgs class (System.Web.UI.WebControls)
2	TrackViewState
3	
4	
5	Description
6	Provides data for the
7	System.Web.UI.WebControls.DataGrid.SortCommand event of the
8	System.Web.UI.WebControls.DataGrid control. This class cannot be inherited.
9	When sorting is enabled by setting the
10	System.Web.UI.WebControls.DataGrid.AllowSorting property of the
11	System.Web.UI.WebControls.DataGrid control to true,
12	System.Web.UI.WebControls.LinkButton controls are rendered in the header of
13	each column that has the
14	System.Web.UI.WebControls.DataGridColumn.SortExpression property set.
15	(For automatically generated columns, the
16	System.Web.UI.WebControls.DataGridColumn.SortExpression property
17	contains the same value as the data field.) These links allow you to sort the
18	System.Web.UI.WebControls.DataGrid control by the selected column. The
19	System.Web.UI.WebControls.DataGrid.SortCommand event is raised when a
20	System.Web.UI.WebControls.LinkButton control is clicked.
21	DataGridSortCommandEventArgs
22	Example Syntax:
23	TrackViewState
24	
25	IC#I public DataGridSortCommandEventArgs(object commandSource.

1	DataGridCommandEventArgs dce);
2	[C++] public: DataGridSortCommandEventArgs(Object* commandSource,
3	DataGridCommandEventArgs* dce);
4	[VB] Public Sub New(ByVal commandSource As Object, ByVal dce As
5	DataGridCommandEventArgs)
6	[JScript] public function DataGridSortCommandEventArgs(commandSource :
7	Object, dce: DataGridCommandEventArgs);
8	
9	Description
10	Initializes a new instance of the
11	System.Web.UI.WebControls.DataGridSortCommandEventArgs class.
12	Use this constructor to create and initialize a new instance of the
13	System.Web.UI.WebControls.DataGridSortCommandEventArgs class. The
14	source of the command. A
15	System.Web.UI.WebControls.DataGridCommandEventArgs that contains the
16	event data.
17	CommandSource
18	TrackViewState
19	
20	[C#] public object CommandSource {get;}
21	[C++] public:property Object* get_CommandSource();
22	[VB] Public ReadOnly Property CommandSource As Object
23	[JScript] public function get CommandSource() : Object;
24	
25	Description

Gets the source of the command. 1 Use the 2 System. Web. UI. Web Controls. Data Grid Sort Command Event Args. Command Sort Command Event Args. Command Sort Command Event Args. Command Sort Command Event Args. Command Sort Command Event Args. Comman3 ource property to get the command source that raises the 4 System.Web.UI.WebControls.DataGrid.SortCommand event. Because this 5 event is only raised when a link is clicked in the header of the System.Web.UI.WebControls.DataGrid control, the returned source is a System.Web.UI.WebControls.DataGridItem that represents the header. This property can be used to programmatically control the header. SortExpression 10 **TrackViewState** 11 12 [C#] public string SortExpression {get;} 13 [C++] public: property String\* get SortExpression(); 14 [VB] Public ReadOnly Property SortExpression As String [JScript] public function get SortExpression(): String; 16 17 Description 18 Gets the expression used to sort the 19 System.Web.UI.WebControls.DataGrid control. 20 Use the 21 System. Web. UI. WebControls. Data Grid Sort Command Event Args. Sort Expres22 sion property to determine which column the user selects to sort the 23 System.Web.UI.WebControls.DataGrid control. 24 25

1	DataGridSortCommandEventHandler delegate
2	(System.Web.UI.WebControls)
3	ToString
4	
5	
6	Description
7	Represents the method that will handle the
8	System.Web.UI.WebControls.DataGrid.SortCommand event of the
9	System.Web.UI.WebControls.DataGrid control. The source of the event. A
10	System.Web.UI.WebControls.DataGridSortCommandEventArgs that contains
11	the event data.
12	When sorting is enabled by setting the
13	System.Web.UI.WebControls.DataGrid.AllowSorting property of the
14	System.Web.UI.WebControls.DataGrid control to true,
15	System.Web.UI.WebControls.LinkButton controls are rendered in the header of
16	each column. These links allow you to sort the
17	System.Web.UI.WebControls.DataGrid control by the selected column. The
18	System.Web.UI.WebControls.DataGrid.SortCommand event is raised when a
19	System.Web.UI.WebControls.LinkButton control is clicked.
20	DataKeyCollection class (System.Web.UI.WebControls)
21	ToString
22	
23	
24	Description
25	

1	Represents a collection of primary key field names. This class cannot be
2	inherited.
3	DataKeyCollection
4	Example Syntax:
5	ToString
6	
7	[C#] public DataKeyCollection(ArrayList keys);
8	[C++] public: DataKeyCollection(ArrayList* keys);
9	[VB] Public Sub New(ByVal keys As ArrayList)
10	[JScript] public function DataKeyCollection(keys : ArrayList);
11	
12	Description
13	Initializes a new instance of the
14	System.Web.UI.WebControls.DataKeyCollection class. A
15	System.Collections.ArrayList to store the keys.
16	Count
17	ToString
18	
19	[C#] public int Count {get;}
20	[C++] public:property int get_Count();
21	[VB] Public ReadOnly Property Count As Integer
22	[JScript] public function get Count(): int;
23	
24	Description
25	Gets the number of objects in the collection. This property is read-only.

1	IsReadOnly
2	ToString
3	
4	[C#] public bool IsReadOnly {get;}
5	[C++] public:property bool get_IsReadOnly();
6	[VB] Public ReadOnly Property IsReadOnly As Boolean
7	[JScript] public function get IsReadOnly(): Boolean;
8	
9	Description
10	Gets the value that specifies whether items in the
11	System.Web.UI.WebControls.DataKeyCollection can be modified. This
12	property is read-only.
13	IsSynchronized
14	ToString
15	
16	[C#] public bool IsSynchronized {get;}
17	[C++] public:property bool get_IsSynchronized();
18	[VB] Public ReadOnly Property IsSynchronized As Boolean
19	[JScript] public function get IsSynchronized(): Boolean;
20	
21	Description
22	Gets a value that indicates whether the
23	System.Web.UI.WebControls.DataKeyCollection is thread-safe. This property
24	is read-only.
25	Item

	1	ToString
	2	
	3	[C#] public object this[int index] {get;}
	4	[C++] public:property Object* get_Item(int index);
	5	[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As Object
	6	[JScript] returnValue = DataKeyCollectionObject.Item(index);
	7	
	8	Description
	9	Gets the primary key field name at the specified index in the collection.
7.5	10	This property is read-only. The index of the primary key field name in the
H., H. M., G.	11	collection to retrieve.
<b>"                              </b>	12	SyncRoot
first first	13	ToString
Henn Stad?	14	
	15	[C#] public object SyncRoot {get;}
	16	[C++] public:property Object* get_SyncRoot();
•	17	[VB] Public ReadOnly Property SyncRoot As Object
	18	[JScript] public function get SyncRoot() : Object;
	19	
	20	Description
	21	Gets the object used to synchronize access to the collection. This property
	22	is read-only.
	23	СоруТо
	24	
	25	[C#] public void CopyTo(Array array, int index);

[C++] public:sealed void CopyTo(Array* array, int index);
[VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
Integer)
[JScript] public function CopyTo(array : Array, index : int);
Description
Copies the contents of the entire collection into an System.Array
appending at the specified index of the System.Array. The System.Array to
copy the contents of the collection into. The index of the System.Array to begin
copying the contents of the collection into.
GetEnumerator
[C#] public IEnumerator GetEnumerator();
[C++] public:sealed IEnumerator* GetEnumerator();
[VB] NotOverridable Public Function GetEnumerator() As IEnumerator
[JScript] public function GetEnumerator() : IEnumerator;
Description
Creates an enumerator for the
System.Web.UI.WebControls.DataKeyCollection used to iterate through the
collection.
DataList class (System.Web.UI.WebControls)
ToString

MS1-863US.APP

Print Brail Print	
7	
2	
į	
death Amil Sant Man,	
1	
1	
Henry	
Ē.	
thank thair	
iluni.	
ì.	

1	
2	
3	Description
4	A data bound list control that displays items using templates.
5	Use the System.Web.UI.WebControls.DataList control to display a
6	template-defined data bound list. The System.Web.UI.WebControls.DataList
7	control supports selecting and editing.
8	ToString
9	
10	[C#] public const string CancelCommandName;
11	[C++] public: const String* CancelCommandName;
12	[VB] Public Const CancelCommandName As String
13	[JScript] public var CancelCommandName : String;
14	
15	Description
16	Represents the Cancel command name. This field is read-only.
17	Use the System.Web.UI.WebControls.DataList.CancelCommandName
18	field to represent the Cancel command name.
19	ToString
20	
21	[C#] public const string DeleteCommandName;
22	[C++] public: const String* DeleteCommandName;
23	[VB] Public Const DeleteCommandName As String
24	[JScript] public var DeleteCommandName : String;

Description 2 Represents the **Delete** command name. This field is read-only. 3 Use the System.Web.UI.WebControls.DataList.DeleteCommandName field to represent the **Delete** command name. 5 **ToString** 6 7 [C#] public const string EditCommandName; 8 [C++] public: const String\* EditCommandName; 9 [VB] Public Const EditCommandName As String 10 [JScript] public var EditCommandName : String; 11 12 Description 13 Represents the **Edit** command name. This field is read-only. 14 Use the System. Web. UI. Web Controls. Data List. Edit Command Name 15 field to represent the Edit command name. 16 **ToString** 17 18 [C#] public const string SelectCommandName; 19 [C++] public: const String\* SelectCommandName; 20 [VB] Public Const SelectCommandName As String 21 [JScript] public var SelectCommandName : String; 22 23 Description 24 Represents the **Select** command name. This field is read-only. 25

	1	Use the System.Web.UI.WebControls.DataList.SelectCommandName
	2	field to represent the Select command name.
	3	ToString
	4	
	5	[C#] public const string UpdateCommandName;
	6	[C++] public: const String* UpdateCommandName;
	7	[VB] Public Const UpdateCommandName As String
	8	[JScript] public var UpdateCommandName : String;
	9	
	10	Description
	11	Represents the <b>Update</b> command name. This field is read-only.
	12	Use the System.Web.UI.WebControls.DataList.UpdateCommandName
	13	field to represent the Update command name.
	14	DataList
4 4	15	Example Syntax:
r hat had be him had	16	ToString
	17	
	18	[C#] public DataList();
	19	[C++] public: DataList();
	20	[VB] Public Sub New()
	21	[JScript] public function DataList();
	22	
	23	Description
	24	Initializes a new instance of the System.Web.UI.WebControls.DataList
	25	class.

Use this constructor to create and initialize a new instance of the 1 System.Web.UI.WebControls.DataList class. 2 AccessKey 3 AlternatingItemStyle **ToString** Description Gets the style properties for alternating items in the 9 System.Web.UI.WebControls.DataList control. 10 Use the System. Web. UI. Web Controls. Data List. Alternating Item Style 11 property to provide a custom style for the alternating items in the 12 System.Web.UI.WebControls.DataList control. Common style attributes that 13 can be adjusted include forecolor, backcolor, font, and content alignment within 14 the cell. Providing a different style enhances the appearance of the 15 System.Web.UI.WebControls.DataList control. AlternatingItemTemplate 17 **ToString** 18 19 [C#] public virtual ITemplate AlternatingItemTemplate {get; set;} 20 [C++] public: property virtual ITemplate\* 21 get AlternatingItemTemplate();public: property virtual void 22 set AlternatingItemTemplate(ITemplate\*); 23 [VB] Overridable Public Property Alternating Item Template As ITemplate 24 [JScript] public function get AlternatingItemTemplate(): ITemplate; public

	1	function set AlternatingItemTemplate(ITemplate);
	2	
	3	Description
	4	Gets or sets the template for alternating items in the
	5	System.Web.UI.WebControls.DataList .
	6	Use the
	7	System.Web.UI.WebControls.DataList.AlternatingItemTemplate property to
	8	control the contents of alternating items in the
	9	System.Web.UI.WebControls.DataList control. The appearance of alternating
	10	items is controlled by the
	11	System.Web.UI.WebControls.DataList.AlternatingItemStyle property.
	12	Attributes
and and	13	BackColor
Me and	14	BorderColor
that that He than the	15	BorderStyle
Arris della	16	BorderWidth
	17	CellPadding
	18	CellSpacing
	19	ChildControlsCreated
	20	ClientID
	21	Context
	22	Controls
	23	ControlStyle
	24	ControlStyleCreated
	25	CssClass

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

DataKeyField
DataKeys
DataKeysArray
DataMember
DataSource
EditItemIndex
ToString

Description

Gets or sets the index number of the selected item in the System.Web.UI.WebControls.DataList control to edit.

Use the **System.Web.UI.WebControls.DataList.EditItemIndex** property to programmatically specify the item selected for editing. You can also use this property to determine the index of the item selected for editing.

EditItemStyle

**ToString** 

[C#] public virtual TableItemStyle EditItemStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get\_EditItemStyle();

[VB] Overridable Public ReadOnly Property EditItemStyle As TableItemStyle

[JScript] public function get EditItemStyle(): TableItemStyle;

Description

Gets the style properties for the item selected for editing in the System.Web.UI.WebControls.DataList control.

Use the System.Web.UI.WebControls.DataList.EditItemStyle property to provide a custom style for the item selected for editing in the System.Web.UI.WebControls.DataList control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the System.Web.UI.WebControls.DataList control.

EditItemTemplate

**ToString** 

[C#] public virtual ITemplate EditItemTemplate {get; set;}
[C++] public: \_\_property virtual ITemplate\* get\_EditItemTemplate();public:
\_\_property virtual void set\_EditItemTemplate(ITemplate\*);
[VB] Overridable Public Property EditItemTemplate As ITemplate
[JScript] public function get EditItemTemplate() : ITemplate;public function set
EditItemTemplate(ITemplate);

Description

Gets or sets the template for the item selected for editing in the System.Web.UI.WebControls.DataList control.

Use the System.Web.UI.WebControls.DataList.EditItemTemplate property to control the contents of the item selected for editing in the System.Web.UI.WebControls.DataList control. The appearance of the item

lee ❷hayes piic 509+324-9256 1789 MS1-863US.APP

selected for editing is controlled by the

 $System. Web. UI. Web Controls. Data List. Edit I tem Style\ property.$ 

Enabled

EnableViewState

**Events** 

ExtractTemplateRows

**ToString** 

## Description

Gets or sets a value that indicates whether the rows of a

System.Web.UI.WebControls.Table control, defined in each template of a

System.Web.UI.WebControls.DataList control, are extracted and displayed.

The contents of the **System.Web.UI.WebControls.DataList** control are specified by using templates. Normally, you list controls that you want to display in the templates. You can also place a **System.Web.UI.WebControls.Table** control in a template and display the rows of the table.

Font

FooterStyle

**ToString** 

### Description

Gets the style properties for the footer section of the **System.Web.UI.WebControls.DataList** control.

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use this property to provide a custom style for the footer section of the System.Web.UI.WebControls.DataList control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the System.Web.UI.WebControls.DataList control.

FooterTemplate

**ToString** 

[C#] public virtual ITemplate FooterTemplate {get; set;}

[C++] public: \_\_property virtual ITemplate\* get\_FooterTemplate();public:

\_\_property virtual void set\_FooterTemplate(ITemplate\*);

[VB] Overridable Public Property FooterTemplate As ITemplate

[JScript] public function get FooterTemplate(): ITemplate; public function set

FooterTemplate(ITemplate);

Description

Gets or sets the template for the footer section of the

System.Web.UI.WebControls.DataList control.

Use the **System.Web.UI.WebControls.DataList.FooterTemplate**property to control the contents of the footer section. The appearance of the footer section is controlled by the **System.Web.UI.WebControls.DataList.FooterStyle** property.

ForeColor

GridLines

**ToString** 

Description

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the grid line style for the

System.Web.UI.WebControls.DataList control when the

System.Web.UI.WebControls.DataList.RepeatLayout property is set to

RepeatLayout.Table.

Use the **System.Web.UI.WebControls.DataList.GridLines** property to specify the grid line style for the **System.Web.UI.WebControls.DataList** control.

The following table lists the possible styles.

HasChildViewState

HeaderStyle

**ToString** 

# Description

Gets the style properties for the heading section of the

System.Web.UI.WebControls.DataList control.

Use this property to provide a custom style for the heading of the System.Web.UI.WebControls.DataList control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the

System.Web.UI.WebControls.DataList control.

HeaderTemplate

**ToString** 

1	
2	[C#] public virtual ITemplate HeaderTemplate {get; set;}
3	[C++] public:property virtual ITemplate* get_HeaderTemplate();public:
4	property virtual void set_HeaderTemplate(ITemplate*);
5	[VB] Overridable Public Property HeaderTemplate As ITemplate
6	[JScript] public function get HeaderTemplate(): ITemplate; public function set
7	HeaderTemplate(ITemplate);
8	
9	Description
10	Gets or sets the template for the heading section of the
11	System.Web.UI.WebControls.DataList control.
12	Use the System.Web.UI.WebControls.DataList.HeaderTemplate
13	property to control the contents of the heading section. The appearance of the
14	header section is controlled by the
15	System.Web.UI.WebControls.DataList.HeaderStyle property.
16	Height
17	HorizontalAlign
18	ID
19	IsTrackingViewState
20	Items
21	ToString
22	
23	
24	Description
25	

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a collection of **System.Web.UI.WebControls.DataListItem** objects representing the individual items within the control.

Use the **System.Web.UI.WebControls.DataList.Items** collection to programmatically control the items in the

System.Web.UI.WebControls.DataList control. The

System.Web.UI.WebControls.DataList.Items collection does not provide any methods to add or remove items to the collection. However, you can control the contents of an item by providing a handler for the

 $System. Web. UI. WebControls. Data List. Item Created \ {\tt event}.$ 

ItemStyle

**ToString** 

[C#] public virtual TableItemStyle ItemStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get\_ItemStyle();

[VB] Overridable Public ReadOnly Property ItemStyle As TableItemStyle

[JScript] public function get ItemStyle(): TableItemStyle;

Description

Gets the style properties for the items in the

System. Web. UI. Web Controls. Data List control.

Use this property to provide a custom style for the items of the System.Web.UI.WebControls.DataList control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the

 ${\bf System. Web. UI. Web Controls. Data List\ control.}$ 

1	ItemTemplate
2	ToString
3	
4	[C#] public virtual ITemplate ItemTemplate {get; set;}
5	[C++] public:property virtual ITemplate* get_ItemTemplate();public:
6	property virtual void set_ItemTemplate(ITemplate*);
7	[VB] Overridable Public Property ItemTemplate As ITemplate
8	[JScript] public function get ItemTemplate(): ITemplate; public function set
9	ItemTemplate(ITemplate);
10	
11	Description
12	Gets or sets the template for the items in the
13	System.Web.UI.WebControls.DataList control.
14	Use the System.Web.UI.WebControls.DataList.ItemTemplate property
15	to control the contents of the items in the
16	System.Web.UI.WebControls.DataList control. The appearance of the items in
17	the System.Web.UI.WebControls.DataList control is controlled by the
18	System.Web.UI.WebControls.DataList.ItemStyle property.
19	NamingContainer
20	Page
21	Parent
22	RepeatColumns
23	ToString
24	
25	

14

15

16

17

18

19

20

21

22

23

24

25

2

Description

Gets or sets the number of columns to display in the **System.Web.UI.WebControls.DataList** control.

Use this property to specify the number of columns that display items in the **System.Web.UI.WebControls.DataList** control. For example, if you set this property to 5, the **System.Web.UI.WebControls.DataList** control displays its items in five columns.

RepeatDirection

**ToString** 

[C#] public virtual RepeatDirection RepeatDirection {get; set;}

[C++] public: \_\_property virtual RepeatDirection get\_RepeatDirection();public:

\_property virtual void set\_RepeatDirection(RepeatDirection);

[VB] Overridable Public Property RepeatDirection As RepeatDirection

[JScript] public function get RepeatDirection(): RepeatDirection; public function set RepeatDirection(RepeatDirection);

Description

Gets or sets whether the **System.Web.UI.WebControls.DataList** control displays vertically or horizontally.

Use the **System.Web.UI.WebControls.DataList.RepeatDirection** property to specify the display direction of the

System.Web.UI.WebControls.DataList control.

1	RepeatLayout
2	ToString
3	
4	[C#] public virtual RepeatLayout RepeatLayout {get; set;}
5	[C++] public:property virtual RepeatLayout get_RepeatLayout();public:
6	property virtual void set_RepeatLayout(RepeatLayout);
7	[VB] Overridable Public Property RepeatLayout As RepeatLayout
8	[JScript] public function get RepeatLayout(): RepeatLayout; public function set
9	RepeatLayout(RepeatLayout);
10	
11	Description
12	Gets or sets whether the control is displayed in a table or flow layout.
13	Use the System.Web.UI.WebControls.DataList.RepeatLayout property
14	to specify whether the items in the System.Web.UI.WebControls.DataList
15	control are displayed in a table. If this property is set to RepeatLayout. Table, the
16	items in the list are displayed in a table. If this property is set to
17	RepeatLayout.Flow, the items in the list are displayed without a table structure.
18	SelectedIndex
19	ToString
20	
21	[C#] public virtual int SelectedIndex {get; set;}
22	[C++] public:property virtual int get_SelectedIndex();public:property virtual
23	<pre>void set_SelectedIndex(int);</pre>
24	[VB] Overridable Public Property SelectedIndex As Integer
25	[JScript] public function get SelectedIndex(): int;public function set

SelectedIndex(int); 2 Description 3 Gets or sets the index of the selected item in the System.Web.UI.WebControls.DataList control. 5 Use the System. Web. UI. Web Controls. Data List. Selected Index property 6 to programmatically specify the selected item in the 7 System.Web.UI.WebControls.DataList control. You can also use this property 8 to determine the index of the selected item. 9 SelectedItem 10 **ToString** 11 12 [C#] public virtual DataListItem SelectedItem {get;} 13 [C++] public: property virtual DataListItem\* get SelectedItem(); 14 [VB] Overridable Public ReadOnly Property SelectedItem As DataListItem 15 [JScript] public function get SelectedItem() : DataListItem; 16 17 Description 18 Gets the selected item in the System.Web.UI.WebControls.DataList 19 control. 20 Use the System. Web. UI. Web Controls. Data List. Selected I tem property to 21 get a System. Web. UI. WebControls. DataListItem object that represents the 22 selected item in the System. Web. UI. Web Controls. DataList control. This object 23 can then be used to access the properties of the selected item. 24 SelectedItemStyle 25

12

13

14

15

16

17

1	ToString
2	
3	[C#] public virt
4	[C++] public: _
5	[VB] Overridab
6	[JScript] public
7	
8	Description
9	Gets the
10	System.Web.Ul

[C#] public virtual TableItemStyle SelectedItemStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get SelectedItemStyle();

[VB] Overridable Public ReadOnly Property SelectedItemStyle As TableItemStyle [JScript] public function get SelectedItemStyle(): TableItemStyle;

Gets the style properties for the selected item in the System.Web.UI.WebControls.DataList control.

Use this property to provide a custom style for the selected item in the System.Web.UI.WebControls.DataList control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within the cell. Providing a different style enhances the appearance of the System.Web.UI.WebControls.DataList control.

SelectedItemTemplate

**ToString** 

18

19

[C#] public virtual ITemplate SelectedItemTemplate {get; set;}

20 | [

 $[C++]\ public: \_\_property\ virtual\ ITemplate*\ get\_SelectedItemTemplate(); public:$ 

21

\_property virtual void set\_SelectedItemTemplate(ITemplate\*);

[VB] Overridable Public Property SelectedItemTemplate As ITemplate

22

[JScript] public function get SelectedItemTemplate(): ITemplate; public function

24

25

 $set\ Selected Item Template (ITemplate);$ 

Description

Gets or sets the template for the selected item in the System.Web.UI.WebControls.DataList control.

Use the **System.Web.UI.WebControls.DataList.SelectedItemTemplate** property to control the contents of the selected item. The appearance of the selected item is controlled by the

System.Web.UI.WebControls.DataList.SelectedItemStyle property.

SeparatorStyle

**ToString** 

[C#] public virtual TableItemStyle SeparatorStyle {get;}

[C++] public: \_\_property virtual TableItemStyle\* get SeparatorStyle();

[VB] Overridable Public ReadOnly Property SeparatorStyle As TableItemStyle [JScript] public function get SeparatorStyle(): TableItemStyle;

Description

Gets the style properties of the separator between each item in the **System.Web.UI.WebControls.DataList** control.

Use the System.Web.UI.WebControls.DataList.SeparatorStyle property to provide a custom style for the separator between each item in the System.Web.UI.WebControls.DataList control. The separator allows you to place an element with custom content between each item in the System.Web.UI.WebControls.DataList control. Common style attributes that can be adjusted include forecolor, backcolor, font, and content alignment within

the cell. Providing a different style enhances the appearance of the

System.Web.UI.WebControls.DataList control.

SeparatorTemplate

ToString

[C#] public virtual ITemplate SeparatorTemplate {get; set;}

[C++] public: \_\_property virtual ITemplate\* get\_SeparatorTemplate();public: \_\_property virtual void set\_SeparatorTemplate(ITemplate\*);

[VB] Overridable Public Property SeparatorTemplate As ITemplate

[JScript] public function get SeparatorTemplate(): ITemplate;public function set SeparatorTemplate(ITemplate);

Description

Cote or sets the template for the graphete between the inverse of the second to the sec

Gets or sets the template for the separator between the items of the System.Web.UI.WebControls.DataList control.

Use the System.Web.UI.WebControls.DataList.SeparatorTemplate property to control the contents of the separator between the items of the System.Web.UI.WebControls.DataList control. The separator allows you to place an element with custom content between each item in the System.Web.UI.WebControls.DataList control. The appearance of the separator between the items of the System.Web.UI.WebControls.DataList control is controlled by the System.Web.UI.WebControls.DataList.SeparatorStyle property.

**ShowFooter** 

**ToString** 

```
[C#] public virtual bool ShowFooter {get; set;}
    [C++] public: __property virtual bool get_ShowFooter();public: __property virtual
 3
     void set ShowFooter(bool);
 4
    [VB] Overridable Public Property ShowFooter As Boolean
 5
    [JScript] public function get ShowFooter(): Boolean; public function set
 6
     ShowFooter(Boolean);
 7
 8
    Description
 9
           Gets or sets a value indicating whether the footer section is displayed in the
10
    System.Web.UI.WebControls.DataList control.
11
           Use the System. Web. UI. WebControls. DataList. Show Footer property to
12
    specify whether the footer section is displayed in the
13
    System.Web.UI.WebControls.DataList control.
14
           ShowHeader
15
           ToString
16
17
    [C#] public virtual bool ShowHeader {get; set;}
18
    [C++] public: property virtual bool get ShowHeader(); public: property
19
    virtual void set ShowHeader(bool);
20
    [VB] Overridable Public Property ShowHeader As Boolean
21
    [JScript] public function get ShowHeader(): Boolean; public function set
22
    ShowHeader(Boolean);
23
24
    Description
25
```

Gets or sets a value indicating whether the header section is displayed in the System.Web.UI.WebControls.DataList control. 2 Use the System.Web.UI.WebControls.DataList.ShowHeader property to 3 specify whether the header section is displayed in the System.Web.UI.WebControls.DataList control. 5 Site 6 Style TabIndex 8 TagKey 9 TagName 10 **TemplateSourceDirectory** 11 ToolTip 12 UniqueID 13 ViewState 14 ViewStateIgnoresCase 15 Visible 16 Width 17 **ToString** 18 19 20 Description 21 Occurs when the Cancel button is clicked for an item in the 22 System.Web.UI.WebControls.DataList control. 23 24

25

The System.Web.UI.WebControls.DataList.CancelCommand event is raised when the Cancel button is clicked for an item in the

System.Web.UI.WebControls.DataList control.

**ToString** 

Description

Occurs when the **Delete** button is clicked for an item in the

System.Web.UI.WebControls.DataList control.

The **System.Web.UI.WebControls.DataList.DeleteCommand** event is raised when the **Delete** button is clicked for an item in the

System.Web.UI.WebControls.DataList control.

**ToString** 

Description

Occurs when the Edit button is clicked for an item in the

System.Web.UI.WebControls.DataList control.

The System.Web.UI.WebControls.DataList.EditCommand event is raised when the Edit button is clicked for an item in the

 ${\bf System. Web. UI. WebControls. Data List\ control.}$ 

**ToString** 

Description

Occurs when any button is clicked in the **System.Web.UI.WebControls.DataList** control.

The System.Web.UI.WebControls.DataList.ItemCommand event is raised when any button is clicked in the System.Web.UI.WebControls.DataList control and is commonly used when you have a button control with a custom CommandName value.

**ToString** 

[C#] public event DataListItemEventHandler ItemCreated;

[C++] public: \_\_event DataListItemEventHandler\* ItemCreated;

[VB] Public Event ItemCreated As DataListItemEventHandler

Description

Occurs on the server when an item in the

System.Web.UI.WebControls.DataList control is created.

The **System.Web.UI.WebControls.DataList.ItemCreated** event is raised when an item in the **System.Web.UI.WebControls.DataList** control is created, both during round-trips and at data bind time.

**ToString** 

[C#] public event DataListItemEventHandler ItemDataBound;

[C++] public: \_\_event DataListItemEventHandler\* ItemDataBound;

[VB] Public Event ItemDataBound As DataListItemEventHandler

Description

Occurs when an item is data bound to the

System.Web.UI.WebControls.DataList control.

The System.Web.UI.WebControls.DataList.ItemDataBound event is raised after an item is data bound to the System.Web.UI.WebControls.DataList control. This event provides you with the last opportunity to access the data item before it is displayed on the client. After this event is raised, the data item is nulled out and no longer available.

**ToString** 

Description

Occurs when the **Update** button is clicked for an item in the **System.Web.UI.WebControls.DataList** control.

The **System.Web.UI.WebControls.DataList.UpdateCommand** event is raised when the **Update** button for an item is clicked.

CreateControlHierarchy

[C#] protected override void CreateControlHierarchy(bool useDataSource);

[C++] protected: void CreateControlHierarchy(bool useDataSource);

[VB] Overrides Protected Sub CreateControlHierarchy(ByVal useDataSource As

Boolean)

21

22

23

24

25

[JScript] protected override function CreateControlHierarchy(useDataSource:

Boolean);

Description

1	CreateControlStyle
2	
3	[C#] protected override Style CreateControlStyle();
4	[C++] protected: Style* CreateControlStyle();
5	[VB] Overrides Protected Function CreateControlStyle() As Style
6	[JScript] protected override function CreateControlStyle(): Style;
7	
8	Description
9	CreateItem
10	
11	[C#] protected virtual DataListItem CreateItem(int itemIndex, ListItemType
12	itemType);
13	[C++] protected: virtual DataListItem* CreateItem(int itemIndex, ListItemType
14	itemType);
15	[VB] Overridable Protected Function CreateItem(ByVal itemIndex As Integer,
16	ByVal itemType As ListItemType) As DataListItem
17	[JScript] protected function CreateItem(itemIndex : int, itemType : ListItemType)
18	: DataListItem;
19	
20	Description
21	InitializeItem
22	
23	[C#] protected virtual void InitializeItem(DataListItem item);
24	[C++] protected: virtual void InitializeItem(DataListItem* item);
25	

1	[VB] Overridable Protected Sub InitializeItem(ByVal item As DataListItem)
2	[JScript] protected function InitializeItem(item : DataListItem);
3	LoadViewState
4	
5	[C#] protected override void LoadViewState(object savedState);
6	[C++] protected: void LoadViewState(Object* savedState);
7	[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)
8	[JScript] protected override function LoadViewState(savedState : Object);
9	
10	Description
11	OnBubbleEvent
12	
13	[C#] protected override bool OnBubbleEvent(object source, EventArgs e);
14	[C++] protected: bool OnBubbleEvent(Object* source, EventArgs* e);
15	[VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,
16	ByVal e As EventArgs) As Boolean
17	[JScript] protected override function OnBubbleEvent(source : Object, e :
18	EventArgs): Boolean;
19	
20	Description
21	OnCancelCommand
22	
23	[C#] protected virtual void OnCancelCommand(DataListCommandEventArgs e)
24	[C++] protected: virtual void OnCancelCommand(DataListCommandEventArgs*
25	e);

1	[VB] Overridable Protected Sub OnCancelCommand(ByVal e As
2	DataListCommandEventArgs)
3	[JScript] protected function OnCancelCommand(e:
4	DataListCommandEventArgs);
5	
6	Description
7	Raises the System.Web.UI.WebControls.DataList.CancelCommand
8	event. This allows you to provide a custom handler for the event.
9	Use the
10	System.Web.UI.WebControls.DataList.OnCancelCommand(System.Web.UI
11	WebControls.DataListCommandEventArgs) method to provide a custom
12	handler for the System.Web.UI.WebControls.DataList.CancelCommand even
13	A System.Web.UI.WebControls.DataListCommandEventArgs that contains
14	event data.
15	OnDeleteCommand
16	
17	[C#] protected virtual void OnDeleteCommand(DataListCommandEventArgs e);
18	[C++] protected: virtual void OnDeleteCommand(DataListCommandEventArgs*
19	e);
20	[VB] Overridable Protected Sub OnDeleteCommand(ByVal e As
21	DataListCommandEventArgs)
22	[JScript] protected function OnDeleteCommand(e:
23	DataListCommandEventArgs);
24	
25	Description

25

Raises the System. Web. UI. Web Controls. Data List. Delete Command event. This allows you to provide a custom handler for the event.

System. Web. UI. WebControls. Data List. On Delete Command (System. Web. UI. System. UI. System. Web. UI. System. Web. UI. System. Web. UI. System. U

WebControls.DataListCommandEventArgs) method to provide a custom

handler for the System. Web.UI. WebControls. DataList. DeleteCommand event.

A System.Web.UI.WebControls.DataListCommandEventArgs that contains

[C#] protected virtual void OnEditCommand(DataListCommandEventArgs e);

[C++] protected: virtual void OnEditCommand(DataListCommandEventArgs\* e);

[VB] Overridable Protected Sub OnEditCommand(ByVal e As

DataListCommandEventArgs)

[JScript] protected function OnEditCommand(e: DataListCommandEventArgs);

Raises the System. Web. UI. Web Controls. Data List. Edit Command event.

This allows you to provide a custom handler for the event.

System. Web. UI. WebControls. Data List. On Edit Command (System. Web. UI. WebControls) and the property ofbControls.DataListCommandEventArgs) method to provide a custom handler for the System.Web.UI.WebControls.DataList.EditCommand event. A System.Web.UI.WebControls.DataListCommandEventArgs that contains event data.

Ο	T4	C	
um	пem	Comm	เลทด

[C#] protected virtual	void On Itam Command	(Data List Common dersont Amas a)	١.
[C#] protected virtual	void UnitemCommand	(DataListCommandEventArgs e)	);

[C++] protected: virtual void OnItemCommand(DataListCommandEventArgs\* e);

[VB] Overridable Protected Sub OnItemCommand(ByVal e As

DataListCommandEventArgs)

[JScript] protected function OnItemCommand(e: DataListCommandEventArgs);

Description

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Raises the **System.Web.UI.WebControls.DataList.ItemCommand** event. This allows you to provide a custom handler for the event.

Use the

System.Web.UI.WebControls.DataList.OnItemCommand(System.Web.UI.W ebControls.DataListCommandEventArgs) method to provide a custom handler for the System.Web.UI.WebControls.DataList.ItemCommand event. A System.Web.UI.WebControls.DataListCommandEventArgs that contains event data.

**OnItemCreated** 

[C#] protected virtual void OnItemCreated(DataListItemEventArgs e);

[C++] protected: virtual void OnItemCreated(DataListItemEventArgs\* e);

[VB] Overridable Protected Sub OnItemCreated(ByVal e As

DataListItemEventArgs)

[JScript] protected function OnItemCreated(e: DataListItemEventArgs);

25

### Description

3

5

7

8

9

10

11

13

17

18

19

20

21

22

23

24

25

Raises the System. Web. UI. Web Controls. Data List. Item Created event.

This allows you to provide a custom handler for the event.

Use the

System.Web.UI.WebControls.DataList.OnItemCreated(System.Web.UI.Web

Controls.DataListItemEventArgs) method to provide a custom handler for the

System.Web.UI.WebControls.DataList.ItemCreated event. A

System.Web.UI.WebControls.DataListItemEventArgs that contains event data.

**OnItemDataBound** 

[C#] protected virtual void OnItemDataBound(DataListItemEventArgs e);

[C++] protected: virtual void OnItemDataBound(DataListItemEventArgs\* e);

[VB] Overridable Protected Sub OnItemDataBound(ByVal e As

DataListItemEventArgs)

[JScript] protected function OnItemDataBound(e: DataListItemEventArgs);

#### Description

Raises the System. Web. UI. Web Controls. Data List. Item Data Bound event. This allows you to provide a custom handler for the event.

Use the

System.Web.UI.WebControls.DataList.OnItemDataBound(System.Web.UI.

WebControls.DataListItemEventArgs) method to provide a custom handler for

the System. Web. UI. Web Controls. Data List. Item Data Bound event. A

System.Web.UI.WebControls.DataListItemEventArgs that contains event data.

On Update Command
-------------------

[C#] protected virtual void OnUpdateCommand(DataListCommandEventArgs e);

[C++] protected: virtual void OnUpdateCommand(DataListCommandEventArgs\*

e);

2

5

6

7

8

9

10

11

12

13

17

18

19

20

21

22

23

24

25

[VB] Overridable Protected Sub OnUpdateCommand(ByVal e As

DataListCommandEventArgs)

[JScript] protected function OnUpdateCommand(e:

DataListCommandEventArgs);

Description

event data.

Raises the **System.Web.UI.WebControls.DataList.UpdateCommand** event. This allows you to provide a custom handler for the event.

Use the

System. Web. UI. Web Controls. Data List. On Update Command (System. Web. UI. System. UI. System. Web. UI.

WebControls.DataListCommandEventArgs) method to provide a custom handler for the System.Web.UI.WebControls.DataList.UpdateCommand

 $event.\ A\ \textbf{System.Web.UI.WebControls.DataListItemEventArgs}\ that\ contains$ 

1813

PrepareControlHierarchy

[C#] protected override void PrepareControlHierarchy();

[C++] protected: void PrepareControlHierarchy();

[VB] Overrides Protected Sub PrepareControlHierarchy()

[JScript] protected override function PrepareControlHierarchy();

1	
2	Description
3	RenderContents
4	
5	[C#] protected override void RenderContents(HtmlTextWriter writer);
6	[C++] protected: void RenderContents(HtmlTextWriter* writer);
7	[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)
8	[JScript] protected override function RenderContents(writer: HtmlTextWriter);
9	
10	Description
11	SaveViewState
12	
13	[C#] protected override object SaveViewState();
14	[C++] protected: Object* SaveViewState();
15	[VB] Overrides Protected Function SaveViewState() As Object
16	[JScript] protected override function SaveViewState(): Object;
17	
18	Description
19	IRepeatInfoUser.GetItemStyle
20	
21	[C#] Style IRepeatInfoUser.GetItemStyle(ListItemType itemType, int
22	repeatIndex);
23	[C++] Style* IRepeatInfoUser::GetItemStyle(ListItemType itemType, int
24	repeatIndex);
25	[VB] Function GetItemStyle(ByVal itemType As ListItemType, ByVal

21

24

25

repeatIndex As Integer) As Style Implements IRepeatInfoUser.GetItemStyle [JScript] function IRepeatInfoUser.GetItemStyle(itemType: ListItemType, repeatIndex : int) : Style; 3 IRepeatInfoUser.RenderItem 4 5 [C#] void IRepeatInfoUser.RenderItem(ListItemType itemType, int repeatIndex, RepeatInfo repeatInfo, HtmlTextWriter writer); 7 [C++] void IRepeatInfoUser::RenderItem(ListItemType itemType, int 8 repeatIndex, RepeatInfo\* repeatInfo, HtmlTextWriter\* writer); 9 [VB] Sub RenderItem(ByVal itemType As ListItemType, ByVal repeatIndex As 10 Integer, ByVal repeatInfo As RepeatInfo, ByVal writer As HtmlTextWriter) 11 Implements IRepeatInfoUser.RenderItem 12 [JScript] function IRepeatInfoUser.RenderItem(itemType: ListItemType, 13 repeatIndex: int, repeatInfo: RepeatInfo, writer: HtmlTextWriter); 14 **TrackViewState** 15 16 [C#] protected override void TrackViewState(); 17 [C++] protected: void TrackViewState(); 18 [VB] Overrides Protected Sub TrackViewState() 19 [JScript] protected override function TrackViewState(); Description 22 Marks the starting point to begin tracking and saving changes to the control 23 as part of the control viewstate.

DataListCommandEventArgs class (System.Web.UI.WebControls)

٠D
1 22
Ľ
ļQ
ijĪ
н
["]
===
i ulk

	1	TrackViewState
	2	
	3	
	4	Description
	5	Provides data for the
	6	$System. Web. UI. Web Controls. Data List. Cancel Command\ ,$
	7	$System. Web. UI. WebControls. Data List. Delete Command\ ,$
	8	$System. Web. UI. Web Controls. Data List. Edit Command\ ,$
	9	${\bf System. Web. UI. WebControls. Data List. Item Command}\ , \ {\bf and}$
	10	System.Web.UI.WebControls.DataList.UpdateCommand events of the
	11	System.Web.UI.WebControls.DataList control. This class cannot be inherited.
	12	The System.Web.UI.WebControls.DataList.CancelCommand event is
	13	raised when the Cancel button for an item in the
- T	14	System.Web.UI.WebControls.DataList control is clicked.
	15	DataListCommandEventArgs
	16	Example Syntax:
ulb	17	TrackViewState
	18	
	19	[C#] public DataListCommandEventArgs(DataListItem item, object
	20	commandSource, CommandEventArgs originalArgs);
	21	[C++] public: DataListCommandEventArgs(DataListItem* item, Object*
	22	commandSource, CommandEventArgs* originalArgs);
	23	[VB] Public Sub New(ByVal item As DataListItem, ByVal commandSource As
	24	Object, ByVal originalArgs As CommandEventArgs)
	25	[JScript] public function DataListCommandEventArgs(item : DataListItem,

commandSource: Object, originalArgs: CommandEventArgs);

Description

2

3

4

6

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the

System.Web.UI.WebControls.DataListCommandEventArgs class.

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.DataListCommandEventArgs class. The selected item from the System.Web.UI.WebControls.DataList. The source of the command. A System.Web.UI.WebControls.CommandEventArgs that contains the original event data.

CommandArgument

CommandName

CommandSource

**TrackViewState** 

Description

Gets the source of the command.

Use the

e property to determine the command source that raised the event. This property is commonly used to determine which command raises the event. You can then take appropriate action, based on the command.

Item

TrackViewState

1	
2	[C#] public DataListItem Item {get;}
3	[C++] public:property DataListItem* get_Item();
4	[VB] Public ReadOnly Property Item As DataListItem
5	[JScript] public function get Item() : DataListItem;
6	
7	Description
8	Gets the item containing the command source in the
9	System.Web.UI.WebControls.DataList control.
10	Use the
11	System.Web.UI.WebControls.DataListCommandEventArgs.Item property to
12	access information about the selected item in the
13	System.Web.UI.WebControls.DataList control.
14	DataListCommandEventHandler delegate (System.Web.UI.WebControls)
15	ToString
16	
17	
18	Description
19	Represents the method that will handle the
20	System.Web.UI.WebControls.DataList.CancelCommand,
21	System.Web.UI.WebControls.DataList.DeleteCommand,
22	System.Web.UI.WebControls.DataList.EditCommand,
23	System.Web.UI.WebControls.DataList.ItemCommand, and
24	System.Web.UI.WebControls.DataList.UpdateCommand events of a
25	System. Web.UI. WebControls. Data List control. The source of the event. A

	- 11	
	2	event data.
	3	The System.Web.UI.WebControls.DataList.CancelCommand event is
	4	raised when the Cancel button for an item in the
	5	System.Web.UI.WebControls.DataList control is clicked.
	6	DataListItem class (System.Web.UI.WebControls)
	7	ToString
	8	
	9	
	10	Description
	11	Represents an item in the System.Web.UI.WebControls.DataList control.
	12	A System.Web.UI.WebControls.DataListItem object represents an item
	13	in the System.Web.UI.WebControls.DataList control, such as the heading
	14	section, the footer section, or a data item.
	15	DataListItem
	16	Example Syntax:
ella.	17	ToString
	18	
	19	[C#] public DataListItem(int itemIndex, ListItemType itemType);
	20	[C++] public: DataListItem(int itemIndex, ListItemType itemType);
	21	[VB] Public Sub New(ByVal itemIndex As Integer, ByVal itemType As
	22	ListItemType)
	23	[JScript] public function DataListItem(itemIndex : int, itemType : ListItemType);
	24	
	25	Description

System.Web.UI.WebControls.DataListCommandEventArgs that contains the

	1	Initializes a new instance of the
	2	System.Web.UI.WebControls.DataListItem class.
	3	Use this constructor to create and initialize a new instance of the
	4	System.Web.UI.WebControls.DataListItem class. The index of the item in the
	5	System.Web.UI.WebControls.DataList control from the
	6	System.Web.UI.WebControls.DataList.Items collection. One of the
	7	System.Web.UI.WebControls.ListItemType values.
	8	AccessKey
	9	Attributes
	10	BackColor
	11	BorderColor
T C	12	BorderStyle
	13	BorderWidth
#1 <u>1</u>	14	ChildControlsCreated
4] A	15	ClientID
	16	Context
222	17	Controls
	18	ControlStyle
	19	ControlStyleCreated
	20	CssClass
	21	DataItem
	22	ToString
	23	
	24	

25 Description

16

17

18

19

20

21

22

23

24

25

1

2

C-4	4	_1 _ 4 _	:4		:41-	41
Gets or	sets a	aata	nem	associated	wiin	ıne

System.Web.UI.WebControls.DataListItem object in the

System.Web.UI.WebControls.DataList control.

Use the **System.Web.UI.WebControls.DataListItem.DataItem** property to specify or determine the properties of a data item associated with the

System.Web.UI.WebControls.DataListItem object in the

System.Web.UI.WebControls.DataList control.

Enabled

EnableViewState

**Events** 

Font

ForeColor

HasChildViewState

Height

ID

IsTrackingViewState

ItemIndex

**ToString** 

Description

Gets the index of the **System.Web.UI.WebControls.DataListItem** object from from the **System.Web.UI.WebControls.DataList.Items** collection of the control.

1	Use the System.Web.UI.WebControls.DataListItem.ItemIndex property
2	to determine the index number of the
3	System.Web.UI.WebControls.DataListItem object from the
4	System.Web.UI.WebControls.DataList.Items collection.
5	ItemType
6	ToString
7	
8	[C#] public virtual ListItemType ItemType {get;}
9	[C++] public:property virtual ListItemType get_ItemType();
10	[VB] Overridable Public ReadOnly Property ItemType As ListItemType
11	[JScript] public function get ItemType() : ListItemType;
12	
13	Description
14	Gets the type of the item represented by the
15	System.Web.UI.WebControls.DataListItem object in the
16	System.Web.UI.WebControls.DataList control.
17	Use the System.Web.UI.WebControls.DataListItem.ItemType property
18	to determine the type of the item represented by the
19	System.Web.UI.WebControls.DataGridItem object. The following table lists
20	the various item types.
21	NamingContainer
22	Page
23	Parent
24	Site
25	Style

1	TabIndex
2	TagKey
3	TagName
4	TemplateSourceDirectory
5	ToolTip
6	UniqueID
7	ViewState
8	ViewStateIgnoresCase
9	Visible
10	Width
11	CreateControlStyle
12	
13	[C#] protected override Style CreateControlStyle();
14	[C++] protected: Style* CreateControlStyle();
15	[VB] Overrides Protected Function CreateControlStyle() As Style
16	[JScript] protected override function CreateControlStyle(): Style;
17	
18	Description
19	OnBubbleEvent
20	
21	[C#] protected override bool OnBubbleEvent(object source, EventArgs e);
22	[C++] protected: bool OnBubbleEvent(Object* source, EventArgs* e);
23	[VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object,
24	ByVal e As EventArgs) As Boolean
25	[JScript] protected override function OnBubbleEvent(source : Object, e :

EventArgs): Boolean; 2 Description 3 RenderItem 4 5 [C#] public virtual void RenderItem(HtmlTextWriter writer, bool extractRows, 6 bool tableLayout); 7 [C++] public: virtual void RenderItem(HtmlTextWriter\* writer, bool extractRows, 8 bool tableLayout); 9 [VB] Overridable Public Sub RenderItem(ByVal writer As HtmlTextWriter, 10 ByVal extractRows As Boolean, ByVal tableLayout As Boolean) 11  $[JScript]\ public\ function\ RenderItem (writer: HtmlTextWriter,\ extractRows:$ 12 Boolean, tableLayout : Boolean); 13 14 Description 15 Displays the System. Web. UI. Web Controls. Data List I tem object on the 16 client. 17 Use the 18 System. Web. UI. WebControls. Data List Item. Render Item (System. Web. UI. Htm.)19 ITextWriter, System. Boolean, System. Boolean) method to display the 20 System.Web.UI.WebControls.DataListItem object on the client. A 21 System. Web. UI. Html Text Writer object that contains the output stream for 22 rendering on the client. true to extract rows; otherwise false. true to display as a 23 table; otherwise false. 24 SetItemType 25

1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	,
18	
19	,

21

22

23

24

array notation.

[C#] protected internal virtual void SetItemType(ListItemType itemType);
[C++] protected public: virtual void SetItemType(ListItemType itemType);
[VB] Overridable Protected Friend Dim Sub SetItemType(ByVal itemType As
ListItemType)
[JScript] package function SetItemType(itemType : ListItemType);
Description
Description C. H. G. H. G. H. W. L. G. H. W. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. L. G. H. W. H. W. L. G. H. W. H. W. L. G. H. W. H. W. L. G. H. W. H. W. L. G. H. W. H. W. L. G. H. W. H. W. L. G. H. W. H. W. L. G. H. W. H.
DataListItemCollection class (System.Web.UI.WebControls)
TrackViewState
Description
Represents the collection of System.Web.UI.WebControls.DataListIte
objects in the System. Web.UI. WebControls. DataList control. This class cann

em ot be inherited.

 $The \ {\bf System. Web. UI. WebControls. Data List Item Collection \ class$ represents a collection of System. Web.UI. WebControls. DataListItem objects, which in turn represent the data items in a System.Web.UI.WebControls.DataList control. To programmatically retrieve System.Web.UI.WebControls.DataListItem objects from a System.Web.UI.WebControls.DataList control, use one of the following methods: Use the indexer to get a single System.Web.UI.WebControls.DataListItem object from the collection, using

1	DataListItemCollection
2	Example Syntax:
3	TrackViewState
4	
5	[C#] public DataListItemCollection(ArrayList items);
6	[C++] public: DataListItemCollection(ArrayList* items);
7	[VB] Public Sub New(ByVal items As ArrayList)
8	[JScript] public function DataListItemCollection(items : ArrayList);
9	
10	Description
11	Initializes a new instance of the
12	System.Web.UI.WebControls.DataListItemCollection class.
13	Use this constructor to create and initialize a new instance of the
14	System.Web.UI.WebControls.DataListItemCollection class. A
15	System.Collections.ArrayList object that contains the items with which to
16	initialize the collection.
17	Count
18	TrackViewState
19	
20	[C#] public int Count {get;}
21	[C++] public:property int get_Count();
22	[VB] Public ReadOnly Property Count As Integer
23	[JScript] public function get Count(): int;
24	
25	Description

1	Gets the number of System.Web.UI.WebControls.DataListItem objects
2	in the collection.
3	Use the System.Web.UI.WebControls.DataListItemCollection.Count
4	property to determine the number of
5	System.Web.UI.WebControls.DataListItem objects in the
6	System.Web.UI.WebControls.DataListItemCollection collection. The
7	System.Web.UI.WebControls.DataListItemCollection.Count property is
8	commonly used when iterating through the collection to determine the upper
9	bound of the collection.
10	IsReadOnly
11	TrackViewState
12	
13	[C#] public bool IsReadOnly {get;}
14	[C++] public:property bool get_IsReadOnly();
15	[VB] Public ReadOnly Property IsReadOnly As Boolean
16	[JScript] public function get IsReadOnly(): Boolean;
17	
18	Description
19	Gets a value that indicates whether the
20	System.Web.UI.WebControls.DataListItem objects in the
21	System.Web.UI.WebControls.DataListItemCollection can be modified.
22	This property always returns false to indicate that the
23	System.Web.UI.WebControls.DataListItemCollection can be written to in all
24	cases.
25	IsSynchronized

1	TrackViewState
2	
3	[C#] public bool IsSynchronized {get;}
4	[C++] public:property bool get_IsSynchronized();
5	[VB] Public ReadOnly Property IsSynchronized As Boolean
6	[JScript] public function get IsSynchronized(): Boolean;
7	
8	Description
9	Gets a value indicating whether access to the
10	System.Web.UI.WebControls.DataListItemCollection is synchronized (thread-
11	safe).
12	This property is derived from the System.Collections.ICollection class and
13	is overridden to always return false.
14	Item
15	TrackViewState
16	
17	[C#] public DataListItem this[int index] {get;}
18	[C++] public:property DataListItem* get_Item(int index);
19	[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As
20	DataListItem
21	[JScript] returnValue = DataListItemCollectionObject.Item(index);
22	
23	Description
24	Gets a System.Web.UI.WebControls.DataListItem object at the specified
25	index in the collection.

Use this indexer to get a System.Web.UI.WebControls.DataListItem
object from the System.Web.UI.WebControls.DataListItemCollection at the
specified index using array notation. The index of the
System. Web. UI. Web Controls. Data List I tem in the collection to retrieve.
SyncRoot
TrackViewState
[C#] public object SyncRoot {get;}
[C++] public:property Object* get_SyncRoot();
[VB] Public ReadOnly Property SyncRoot As Object
[JScript] public function get SyncRoot(): Object;
Description
Gets the object that can be used to synchronize access to the
System.Web.UI.WebControls.DataListItemCollection collection.
The object returned in this implementation is the
System.Web.UI.WebControls.DataListItemCollection object itself.
СоруТо
[C#] public void CopyTo(Array array, int index);
[C++] public:sealed void CopyTo(Array* array, int index);
[VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
Integer)
[JScript] public function CopyTo(array: Array, index: int);

Description

Copies all the items from this

System.Web.UI.WebControls.DataListItemCollection collection to the specified System.Array object, starting at the specified index in the System.Array object.

Use this method to copy the contents of the

System.Web.UI.WebControls.DataListItemCollection collection into the specified System.Array object, starting at the specified index. A zero-based System.Array object that receives the copied items from the System.Web.UI.WebControls.DataListItemCollection collection. The first position in the specified System.Array object to receive the copied contents.

GetEnumerator

[C#] public IEnumerator GetEnumerator();

[C++] public: sealed IEnumerator\* GetEnumerator();

[VB] NotOverridable Public Function GetEnumerator() As IEnumerator

[JScript] public function GetEnumerator(): IEnumerator;

Description

Returns a System.Collections.IEnumerator interface that contains all System.Web.UI.WebControls.DataListItem objects in the System.Web.UI.WebControls.DataListItemCollection .

Return Value: A System.Collections.IEnumerator interface that contains all

lee@hayes pik 509-324-9256 1830 MS1-863US.APP

1	System.Web.UI.WebControls.DataListItem objects in the
2	$System. Web. UI. Web Controls. Data List I tem Collection \ .$
3	Use this method to create a System.Collections.IEnumerator that can be
4	easily iterated through to get each item in the
5	System.Web.UI.WebControls.DataListItemCollection collection.
6	DataListItemEventArgs class (System.Web.UI.WebControls)
7	ToString
8	
9	
10	Description
11	Provides data for the
12	System.Web.UI.WebControls.DataList.ItemCreated and
13	System.Web.UI.WebControls.DataList.ItemDataBound events of a
14	System.Web.UI.WebControls.DataList control. This class cannot be inherited.
15	The System.Web.UI.WebControls.DataList.ItemCreated event is raised
16	when an item in the System. Web. UI. Web Controls. DataList control is created.
17	DataListItemEventArgs
18	Example Syntax:
19	ToString
20	
21	[C#] public DataListItemEventArgs(DataListItem item);
22	[C++] public: DataListItemEventArgs(DataListItem* item);
23	[VB] Public Sub New(ByVal item As DataListItem)
24	[JScript] public function DataListItemEventArgs(item: DataListItem);
25	

Description

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Initializes a new instance of the

 $System. Web. UI. Web Controls. Data List I tem Event Args \ class.$ 

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.DataListItemEventArgs class. A System.Web.UI.WebControls.DataListItem object that represents an item in the System.Web.UI.WebControls.DataList control.

Item

**ToString** 

[C#] public DataListItem Item {get;}

[C++] public: property DataListItem\* get\_Item();

[VB] Public ReadOnly Property Item As DataListItem

[JScript] public function get Item(): DataListItem;

Description

Gets the referenced item in the **System.Web.UI.WebControls.DataList** control when the event is raised.

The items in the **System.Web.UI.WebControls.DataList** control are referenced as each item is created or bound to the control. Use this property to programmatically access the referenced item in the

System.Web.UI.WebControls.DataList control when the

System.Web.UI.WebControls.DataList.ItemCreated or

System.Web.UI.WebControls.DataList.ItemDataBound event is raised.

DataListItemEventHandler delegate (System.Web.UI.WebControls)
ToString
Description
Represents the method that will handle the
System.Web.UI.WebControls.DataList.ItemCreated and
System.Web.UI.WebControls.DataList.ItemDataBound events of the
System.Web.UI.WebControls.DataList control. The source of the event. A
System.Web.UI.WebControls.DataListItemEventArgs that contains the event
data.
The System.Web.UI.WebControls.DataList.ItemCreated event is raised
when an item in the System. Web. UI. Web Controls. Data List control is created.
DayNameFormat enumeration (System.Web.UI.WebControls)
ToString
Description
Specifies the display format for the days of the week on a
System.Web.UI.WebControls.Calendar control.
The System.Web.UI.WebControls.DayNameFormat enumeration
represents the display formats for the days of the week on a
System.Web.UI.WebControls.Calendar control.
ToString

C#] public const DayNameFormat FirstLetter;
C++] public: const DayNameFormat FirstLetter;
VB] Public Const FirstLetter As DayNameFormat
JScript] public var FirstLetter : DayNameFormat;
Description
The days of the week displayed with just the first letter. For example, T.
ToString
[C#] public const DayNameFormat FirstTwoLetters;
[C++] public: const DayNameFormat FirstTwoLetters;
[VB] Public Const FirstTwoLetters As DayNameFormat
[JScript] public var FirstTwoLetters : DayNameFormat;
Description
The days of the week displayed with just the first two letters. For example,
Ги.
ToString
[C#] public const DayNameFormat Full;
[C++] public: const DayNameFormat Full;
[VB] Public Const Full As DayNameFormat
[JScript] public var Full : DayNameFormat;

Description
The days of the week displayed in full format. For example, Tuesday.
ToString
[C#] public const DayNameFormat Short;
[C++] public: const DayNameFormat Short;
[VB] Public Const Short As DayNameFormat
[JScript] public var Short : DayNameFormat;
Description
The days of the week displayed in abbreviated format. For example, Tues.
DayRenderEventArgs class (System.Web.UI.WebControls)
ToString
Description
Provides data for the
System.Web.UI.WebControls.Calendar.DayRender event of the
System.Web.UI.WebControls.Calendar control. This class cannot be inherited.
Although data binding is not supported for the
System.Web.UI.WebControls.Calendar control, it is possible to modify the
content and formatting of the individual date cells. Before the
System.Web.UI.WebControls.Calendar control is displayed on the Web page, it
creates and assembles the components that make up the control. The

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Cell

System.Web.UI.WebControls.Calendar.DayRender event is raised when each date cell in System. Web. UI. Web Controls. Calendar control is created. You can control the contents and formatting of a date cell when it is created by providing code in the event handler for the System.Web.UI.WebControls.Calendar.DayRender event. DayRenderEventArgs Example Syntax: **ToString** [C#] public DayRenderEventArgs(TableCell cell, CalendarDay day); [C++] public: DayRenderEventArgs(TableCell\* cell, CalendarDay\* day); [VB] Public Sub New(ByVal cell As TableCell, ByVal day As CalendarDay) [JScript] public function DayRenderEventArgs(cell: TableCell, day: CalendarDay); Description Initializes a new instance of the System.Web.UI.WebControls.DayRenderEventArgs class. Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.DayRenderEventArgs class. A System.Web.UI.WebControls.TableCell object that represents a cell in the System.Web.UI.WebControls.Calendar control. A System.Web.UI.WebControls.CalendarDay object that represents the day to render in the System. Web. UI. Web Controls. Calendar control.

1	ToString
2	
3	[C#] public TableCell Cell {get;}
4	[C++] public:property TableCell* get_Cell();
5	[VB] Public ReadOnly Property Cell As TableCell
6	[JScript] public function get Cell(): TableCell;
7	
8	Description
9	Gets the System.Web.UI.WebControls.TableCell object that represents
10	the cell being rendered in the System.Web.UI.WebControls.Calendar control.
11	Use the System. Web. UI. Web Controls. Day Render Event Args. Cell
12	property to programmatically control the cell being rendered in the
13	System.Web.UI.WebControls.Calendar control.
14	Day
15	ToString
16	
17	[C#] public CalendarDay Day {get;}
18	[C++] public:property CalendarDay* get_Day();
19	[VB] Public ReadOnly Property Day As CalendarDay
20	[JScript] public function get Day(): CalendarDay;
21	
22	Description
23	Gets the System.Web.UI.WebControls.CalendarDay that represents the
24	day being rendered in the System.Web.UI.WebControls.Calendar control.

Use the System.Web.UI.WebControls.DayRenderEventArgs.Day property to get information about the day being rendered in the System.Web.UI.WebControls.Calendar control.

DayRenderEventHandler delegate (System.Web.UI.WebControls)
ToString

## Description

Represents the method that will handle the

System.Web.UI.WebControls.Calendar.DayRender event of the

System.Web.UI.WebControls.Calendar control. The source of the event. A

System.Web.UI.WebControls.DayRenderEventArgs that contains the event data.

Although data binding is not supported for the

System.Web.UI.WebControls.Calendar control, it is possible to modify the
content and formatting of the individual date cells. Before the

System.Web.UI.WebControls.Calendar control is displayed on the Web page, it
creates and assembles the components that make up the control. The

System.Web.UI.WebControls.Calendar.DayRender event is raised when each
date cell in System.Web.UI.WebControls.Calendar control is created. You can
control the contents and formatting of a date cell when it is created by providing
code in the event handler for the

System.Web.UI.WebControls.Calendar.DayRender event. For additional

lee@hayes plk 509-324-9256 1838 MS1-863US.APP

information on customizing the contents of a date cell, see

1	System.Web.UI.WebControls.Calendar.OnDayRender(System.Web.UI.Web
2	$Controls. Table Cell, System. Web. UI. Web Controls. Calendar Day)\ .$
3	DropDownList class (System.Web.UI.WebControls)
4	ToString
5	
6	
7	Description
8	Represents a control that allows the user to select a single item from a drop-
9	down list.
 10	Use the System.Web.UI.WebControls.DropDownList control to create a
11	single selection drop-down list control. You can control the appearance of the
12	System.Web.UI.WebControls.DropDownList control by setting the
13	System.Web.UI.WebControls.DropDownList.BorderColor,
14	System.Web.UI.WebControls.DropDownList.BorderStyle, and
15	System.Web.UI.WebControls.DropDownList.BorderWidth properties.
16	DropDownList
17	Example Syntax:
18	ToString
19	
20	[C#] public DropDownList();
21	[C++] public: DropDownList();
22	[VB] Public Sub New()
23	[JScript] public function DropDownList();
24	
25	Description

3

5

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Initializes a new instance of the

System.Web.UI.WebControls.DropDownList class.

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.DropDownList class.

AccessKey

Attributes

AutoPostBack

BackColor

BorderColor

**ToString** 

## Description

Gets or sets the border color of the control.

The System.Web.UI.WebControls.DropDownList.BorderColor property is inherited from the System.Web.UI.WebControls.WebControl class and is not applicable to the System.Web.UI.WebControls.DropDownList control.

BorderStyle

**ToString** 

[C#] public override BorderStyle BorderStyle {get; set;}

 $[C++]\ public: \underline{\hspace{1.5cm}} property\ virtual\ BorderStyle\ get\_BorderStyle(); public:$ 

\_\_property virtual void set\_BorderStyle(BorderStyle);

[VB] Overrides Public Property BorderStyle As BorderStyle

[JScript] public function get BorderStyle(): BorderStyle; public function set

BorderStyle(BorderStyle);

Description

Gets or sets the border style of the control.

The System.Web.UI.WebControls.DropDownList.BorderStyle property is inherited from the System.Web.UI.WebControls.WebControl class and is not applicable to the System.Web.UI.WebControls.DropDownList control.

**BorderWidth** 

**ToString** 

[C#] public override Unit BorderWidth {get; set;}

[C++] public: \_\_property virtual Unit get\_BorderWidth();public: \_\_property virtual void set BorderWidth(Unit);

[VB] Overrides Public Property BorderWidth As Unit

[JScript] public function get BorderWidth(): Unit;public function set BorderWidth(Unit);

Description

Gets or sets the border width for the control.

The System.Web.UI.WebControls.DropDownList.BorderWidth property is inherited from the System.Web.UI.WebControls.WebControl class and is not applicable to the System.Web.UI.WebControls.DropDownList control.

ChildControlsCreated

1	ClientID
2	Context
3	Controls
4	ControlStyle
5	ControlStyleCreated
6	CssClass
7	DataMember
8	DataSource
9	DataTextField
10	DataTextFormatString
11	DataValueField
12	Enabled
13	EnableViewState
14	Events
15	Font
16	ForeColor
17	HasChildViewState
18	Height
19	ID
20	IsTrackingViewState
21	Items
22	NamingContainer
23	Page
24	Parent

SelectedIndex

**ToString** 

2

3

5

6

7

8

9

10

13

12

14 15

16

17

18 19

20

21

23

24

25 th

Description

Gets or sets the index of the selected item in the System. Web.UI. WebControls. DropDownList control.

Use the System.Web.UI.WebControls.DropDownList.SelectedIndex property to programmatically specify or determine the index of the selected item from the System.Web.UI.WebControls.DropDownList control. An item is always selected in the System.Web.UI.WebControls.DropDownList control.

You cannot deselect every item in the list at the same time.

SelectedItem

Site

Style

TabIndex

TagKey

TagName

TemplateSourceDirectory

ToolTip

**ToString** 

Description

Gets or sets the ToolTip text displayed when the mouse pointer rests over the control.

3

5

6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

The System.Web.UI.WebControls.DropDownList.ToolTip property is inherited from the System. Web.UI. WebControls. WebControl class and is not applicable to the System. Web.UI. WebControls. DropDownList control. UniqueID ViewState ViewStateIgnoresCase Visible Width AddAttributesToRender [C#] protected override void AddAttributesToRender(HtmlTextWriter writer); [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer); [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As HtmlTextWriter) [JScript] protected override function AddAttributesToRender(writer: HtmlTextWriter); Description Adds the properties of the System. Web.UI. WebControls. DropDownList control to the output stream for rendering on the client. The output stream for rendering on the client. CreateControlCollection [C#] protected override ControlCollection CreateControlCollection();  $[C++]\ protected:\ Control Collection *\ Create Control Collection ();$ 

1	[VB] Overrides Protected Function CreateControlCollection() As
2	ControlCollection
3	[JScript] protected override function CreateControlCollection():
4	ControlCollection;
5	
6	Description
7	
8	RenderContents
9	
10	[C#] protected override void RenderContents(HtmlTextWriter writer);
11	[C++] protected: void RenderContents(HtmlTextWriter* writer);
12	[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)
13	[JScript] protected override function RenderContents(writer: HtmlTextWriter);
14	
15	Description
16	Displays the System.Web.UI.WebControls.DropDownList control on the
17	client. The output stream for rendering on the client.
18	IPostBackDataHandler.LoadPostData
19	
20	[C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
21	NameValueCollection postCollection);
22	[C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
23	NameValueCollection* postCollection);
24	[VB] Function LoadPostData(ByVal postDataKey As String, ByVal
25	postCollection As NameValueCollection) As Boolean Implements

24

25

IPostBackDataHandler.LoadPostData [JScript] function IPostBackDataHandler.LoadPostData(postDataKey: String, 2 postCollection: NameValueCollection): Boolean; 3 IPostBackDataHandler. RaisePostDataChangedEvent4 5 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent(); 6 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent(); 7 [VB] Sub RaisePostDataChangedEvent() Implements 8 IPostBackDataHandler. RaisePostDataChangedEvent9 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent(); 10 EditCommandColumn class (System.Web.UI.WebControls) 11 **TrackViewState** 12 13 14 Description 15 A special column type for the System.Web.UI.WebControls.DataGrid 16 control that contains the Edit command buttons for editing data items in each row. 17 Use the System. Web. UI. WebControls. Edit Command Column class to 18 create a special column for the System. Web. UI. Web Controls. Data Grid control 19 that contains the Edit, Update, and Cancel command buttons for each data row 20 in the grid. These buttons allow you to edit the values of a row in the 21  $System. Web. UI. Web Controls. Data Grid\ control.$ 22

EditCommandColumn

Example Syntax:

TrackViewState

1	
2	[C#] public EditCommandColumn();
3	[C++] public: EditCommandColumn();
4	[VB] Public Sub New()
5	[JScript] public function EditCommandColumn();
6	
7	Description
8	Initializes a new instance of the
9	System.Web.UI.WebControls.EditCommandColumn class.
10	Use this constructor to create and initialize a new instance of the
11	System.Web.UI.WebControls.EditCommandColumn class.
12	ButtonType
13	TrackViewState
14	
15	[C#] public virtual ButtonColumnType ButtonType {get; set;}
16	[C++] public:property virtual ButtonColumnType get_ButtonType();public:
17	property virtual void set_ButtonType(ButtonColumnType);
18	[VB] Overridable Public Property ButtonType As ButtonColumnType
19	[JScript] public function get ButtonType(): ButtonColumnType;public function
. 20	set ButtonType(ButtonColumnType);
21	
22	Description
23	Gets or sets the button type for the column.
24	
25	

1	Use the
2	System.Web.UI.WebControls.EditCommandColumn.ButtonType property to
3	specify whether the buttons in the column display as push buttons or hyperlinks.
4	CancelText
5	TrackViewState
6	
7	[C#] public virtual string CancelText {get; set;}
8	[C++] public:property virtual String* get_CancelText();public:property
9	virtual void set_CancelText(String*);
10	[VB] Overridable Public Property CancelText As String
11	[JScript] public function get CancelText(): String; public function set
12	CancelText(String);
13	
14	Description
15	Gets or sets the text to display for the Cancel command button in the
16	System.Web.UI.WebControls.EditCommandColumn .
17	Use the
18	System.Web.UI.WebControls.EditCommandColumn.CancelText property to
19	specify the text to display for the Cancel command button in the
20	System.Web.UI.WebControls.EditCommandColumn .
21	DesignMode
22	EditText
23	TrackViewState
24	
25	

Description

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the text to display for the Edit command button in the  $System. Web. UI. Web Controls. Edit Command Column \ .$ 

 $Use \ the \ \textbf{System.Web.UI.WebControls.} \\ \textbf{EditCommandColumn.EditText}$ property to specify the text to display for the Edit command button in the  $System. Web. UI. Web Controls. Edit Command Column \ .$ 

FooterStyle

**FooterText** 

HeaderImageUrl

HeaderStyle

HeaderText

IsTrackingViewState

ItemStyle

Owner

SortExpression

UpdateText

**TrackViewState** 

Description

Gets or sets the text to display for the Update command button in the  $System. Web. UI. Web Controls. Edit Command Column \ .$ 

1	Use the
2	System.Web.UI.WebControls.EditCommandColumn.UpdateText property to
3	specify the text to display for the Update command button in the
4	$System. Web. UI. Web Controls. Edit Command Column \ .$
5	ViewState
6	Visible
7	InitializeCell
8	
9	[C#] public override void InitializeCell(TableCell cell, int columnIndex,
10	ListItemType itemType);
11	[C++] public: void InitializeCell(TableCell* cell, int columnIndex, ListItemType
12	itemType);
13	[VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal
14	columnIndex As Integer, ByVal itemType As ListItemType)
15	[JScript] public override function InitializeCell(cell: TableCell, columnIndex: int
16	itemType : ListItemType);
17	
18	Description
19	Initializes a cell within the column. A
20	System.Web.UI.WebControls.TableCell that contains information about the cell
21	to initialize. The column number where the cell is located. One of the
22	System.Web.UI.WebControls.ListItemType values.
23	FirstDayOfWeek enumeration (System.Web.UI.WebControls)
24	TrackViewState
25	

	6
	7
	8
	9
full flui	10
	11
Tool Trees	12
shulf stad: shall shum shalf shafe stadi shalf	13
	14
ાં પ્રઅર્થ મેતાનં મંં તેવાત પ્રતાસ	
	15
	16

18

19

20

21

22

23

24

Description

2

3

4

5

Specifies the day to display as the first day of the week on the **System.Web.UI.WebControls.Calendar** control.

The **System.Web.UI.WebControls.FirstDayOfWeek** enumeration represents the values that specify which day to display as the first day of the week on the **System.Web.UI.WebControls.Calendar** control.

TrackViewState

[C#] public const FirstDayOfWeek Default;

[C++] public: const FirstDayOfWeek Default;

[VB] Public Const Default As FirstDayOfWeek

[JScript] public var Default : FirstDayOfWeek;

Description

The first day of the week is specified by the system settings.

**TrackViewState** 

[C#] public const FirstDayOfWeek Friday;

[C++] public: const FirstDayOfWeek Friday;

[VB] Public Const Friday As FirstDayOfWeek

[JScript] public var Friday : FirstDayOfWeek;

Description

1	The first day of the week is Friday.
2	TrackViewState
3	
4	[C#] public const FirstDayOfWeek Monday;
5	[C++] public: const FirstDayOfWeek Monday;
6	[VB] Public Const Monday As FirstDayOfWeek
7	[JScript] public var Monday : FirstDayOfWeek;
8	
9	Description
10	The first day of the week is Monday.
11	TrackViewState
12	
13	[C#] public const FirstDayOfWeek Saturday;
14	[C++] public: const FirstDayOfWeek Saturday;
15	[VB] Public Const Saturday As FirstDayOfWeek
16	[JScript] public var Saturday : FirstDayOfWeek;
17	
18	Description
19	The first day of the week is Saturday.
20	TrackViewState
21	
22	[C#] public const FirstDayOfWeek Sunday;
23	[C++] public: const FirstDayOfWeek Sunday;
24	[VB] Public Const Sunday As FirstDayOfWeek
25	[JScript] public var Sunday : FirstDayOfWeek;

1	
2	Description
3	The first day of the week is Sunday.
4	TrackViewState
5	
6	[C#] public const FirstDayOfWeek Thursday;
7	[C++] public: const FirstDayOfWeek Thursday;
8	[VB] Public Const Thursday As FirstDayOfWeek
9	[JScript] public var Thursday: FirstDayOfWeek;
10	
11	Description
12	The first day of the week is Thursday.
13	TrackViewState
14	
15	[C#] public const FirstDayOfWeek Tuesday;
16	[C++] public: const FirstDayOfWeek Tuesday;
17	[VB] Public Const Tuesday As FirstDayOfWeek
18	[JScript] public var Tuesday : FirstDayOfWeek;
19	
20	Description
21	The first day of the week is Tuesday.
22	TrackViewState
23	
24	[C#] public const FirstDayOfWeek Wednesday;
25	[C++] public: const FirstDayOfWeek Wednesday;

1	[VB] Public Const Wednesday As FirstDayOfWeek
2	[JScript] public var Wednesday : FirstDayOfWeek;
3	
4	Description
5	The first day of the week is Wednesday.
6	FontInfo class (System.Web.UI.WebControls)
7	ToString
8	
9	
10	Description
11	Encapsulates the font properties of text. This class cannot be inherited.
12	Use the System. Web. UI. Web Controls. Font Info class to encapsulate the
13	font properties of text. You can specify the font name and the font size. You can
14	also specify whether the style of the font is bold, italic, overlined, strikethrough, or
15	underlined.
16	Bold
17	ToString
18	
19	[C#] public bool Bold {get; set;}
20	[C++] public:property bool get_Bold();public:property void set_Bold(bool);
21	[VB] Public Property Bold As Boolean
22	[JScript] public function get Bold(): Boolean; public function set Bold(Boolean);
23	
24	Description
25	Gets or sets a value that indicates whether the font is bold.

1	Use the System.Web.UI.WebControls.FontInfo.Bold property to specify
2	or determine whether the font is bold.
3	Italic
4	ToString
5	
6	[C#] public bool Italic {get; set;}
7	[C++] public:property bool get_Italic();public:property void set_Italic(bool);
8	[VB] Public Property Italic As Boolean
9	[JScript] public function get Italic(): Boolean; public function set Italic(Boolean);
10	
11	Description
12	Gets or sets a value that indicates whether the font is italic.
13	Use the System.Web.UI.WebControls.FontInfo.Italic property to specify
14	or determine whether the font is italic.
15	Name
16	ToString
17	
18	[C#] public string Name {get; set;}
19	[C++] public:property String* get_Name();public:property void
20	set_Name(String*);
21	[VB] Public Property Name As String
22	[JScript] public function get Name(): String; public function set Name(String);
23	
24	Description
25	Gets or sets the primary font name.

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

[C#] public bool Overline {get; set;}

Use the System.Web.UI.WebControls.FontInfo.Name property to specify or determine the primary font name. The primary font name determines the font that is used to display text in the control associated with the System. Web. UI. Web Controls. Font Info.Names **ToString** [C#] public string[] Names {get; set;} [C++] public: \_\_property String\* get\_Names();public: \_\_property void set Names(String\* gc[]); [VB] Public Property Names As String () [JScript] public function get Names(): String[];public function set Names(String[]); Description Gets or sets an ordered array of font names. Use the System. Web. UI. Web Controls. Font Info. Names property to specify or determine an ordered array of font names for a  $System. Web. UI. Web Controls. Font Info. \ The$ System.Web.UI.WebControls.FontInfo.Names property is commonly used to store a list of available font names. Overline **ToString** 

[C++] public:property bool get_Overline();public:property void
set_Overline(bool);
[VB] Public Property Overline As Boolean
[JScript] public function get Overline(): Boolean; public function set
Overline(Boolean);
Overmic (Boolean),
D. Sadan
Description
Gets or sets a value that indicates whether the font is overlined.
Use the System.Web.UI.WebControls.FontInfo.Overline property to
specify or determine whether the font contains a line above the text.
Size
ToString
[C#] public FontUnit Size {get; set;}
[C++] public:property FontUnit get_Size();public:property void
[C++] public:property FontUnit get_Size();public:property void
[C++] public:property FontUnit get_Size();public:property void set_Size(FontUnit);
[C++] public:property FontUnit get_Size();public:property void  set_Size(FontUnit);  [VB] Public Property Size As FontUnit
[C++] public:property FontUnit get_Size();public:property void  set_Size(FontUnit);  [VB] Public Property Size As FontUnit
[C++] public:property FontUnit get_Size();public:property void  set_Size(FontUnit);  [VB] Public Property Size As FontUnit  [JScript] public function get Size() : FontUnit;public function set Size(FontUnit);
[C++] public:property FontUnit get_Size();public:property void  set_Size(FontUnit);  [VB] Public Property Size As FontUnit  [JScript] public function get Size() : FontUnit;public function set Size(FontUnit);  Description
[C++] public:property FontUnit get_Size();public:property void  set_Size(FontUnit);  [VB] Public Property Size As FontUnit  [JScript] public function get Size() : FontUnit;public function set Size(FontUnit);  Description  Gets or sets the font size.
[C++] public:property FontUnit get_Size();public:property void  set_Size(FontUnit);  [VB] Public Property Size As FontUnit  [JScript] public function get Size() : FontUnit;public function set Size(FontUnit);  Description  Gets or sets the font size.  Use the System.Web.UI.WebControls.FontInfo.Size property to specify

1	
2	[C#] public bool Strikeout {get; set;}
3	[C++] public:property bool get_Strikeout();public:property void
4	set_Strikeout(bool);
5	[VB] Public Property Strikeout As Boolean
6	[JScript] public function get Strikeout(): Boolean; public function set
7	Strikeout(Boolean);
8	
9	Description
10	Gets or sets a value that indicates whether the font is strikethrough.
11	Use the System.Web.UI.WebControls.FontInfo.Strikeout property to
12	specify or determine whether the font contains a line through the text.
13	Underline
14	ToString
15	
16	[C#] public bool Underline {get; set;}
17	[C++] public:property bool get_Underline();public:property void
18	set_Underline(bool);
19	[VB] Public Property Underline As Boolean
20	[JScript] public function get Underline(): Boolean; public function set
21	Underline(Boolean);
22	
23	Description
24	Gets or sets a value that indicates whether the font is underlined.
25	

Use the System.Web.UI.WebControls.FontInfo.Underline property to 1 specify or determine whether the font contains a line under the text. CopyFrom 3 [C#] public void CopyFrom(FontInfo f); 5 [C++] public: void CopyFrom(FontInfo\* f); [VB] Public Sub CopyFrom(ByVal f As FontInfo) [JScript] public function CopyFrom(f: FontInfo); 9 Description 10 Duplicates the font properties of the specified 11 System.Web.UI.WebControls.FontInfo into the current instance of the 12 System.Web.UI.WebControls.FontInfo class. 13 Use the 14 System.Web.UI.WebControls.FontInfo.CopyFrom(System.Web.UI.WebCont 15 rols.FontInfo) method to duplicate the font properties of the specified 16 System.Web.UI.WebControls.FontInfo into the current instance of the 17 System.Web.UI.WebControls.FontInfo class. A 18 System.Web.UI.WebControls.FontInfo that contains the font properties to 19 duplicate. 20 MergeWith 21 22 [C#] public void MergeWith(FontInfo f); 23 [C++] public: void MergeWith(FontInfo\* f); 24 [VB] Public Sub MergeWith(ByVal f As FontInfo) 25

1	[JScript] public function MergeWith(f: FontInfo);
2	
3	Description
4	Combines the font properties of the specified
5	System.Web.UI.WebControls.FontInfo with the current instance of the
6	System.Web.UI.WebControls.FontInfo class.
7	Use the
8	System.Web.UI.WebControls.FontInfo.MergeWith(System.Web.UI.WebCont
9	rols.FontInfo) method to combine the font properties of the specified
10	System.Web.UI.WebControls.FontInfo with the current instance of the
11	System.Web.UI.WebControls.FontInfo class. A
12	System.Web.UI.WebControls.FontInfo that contains the font properties to
13	combine.
14	ShouldSerializeNames
15	
16	[C#] public bool ShouldSerializeNames();
17	[C++] public: bool ShouldSerializeNames();
18	[VB] Public Function ShouldSerializeNames() As Boolean
19	[JScript] public function ShouldSerializeNames(): Boolean;
20	
21	Description
22	Determines whether the System.Web.UI.WebControls.FontInfo.Names
23	property contains enough entries to serialize.
24	Return Value: true if the System.Web.UI.WebControls.FontInfo.Names
25	property contains enough entries to serialize; otherwise, false.

1 Use the System. Web. UI. Web Controls. Font Info. Should Serialize Names method to determine whether the System. Web. UI. Web Controls. Font Info. Names property contains enough entries 3 to serialize. **ToString** 5 [C#] public override string ToString(); [C++] public: String\* ToString(); [VB] Overrides Public Function ToString() As String [JScript] public override function ToString(): String; 10 11 Description 12 Returns a string that contains the font name and size for an instance of the 13 System.Web.UI.WebControls.FontInfo class. 14 Return Value: A string that contains the font name and size for an instance of the 15 System.Web.UI.WebControls.FontInfo class. 16 Use the System. Web. UI. Web Controls. Font Info. To String method to 17 create a string that contains the font name and size, separated by a comma, for an 18 instance of the System. Web. UI. WebControls. FontInfo class. 19 FontNamesConverter class (System.Web.UI.WebControls) 20 **ToString** 21 22 23 Description 24

21

22

23

24

1

2

3

5

7

Converts a string containing a list of font names to an array of strings containing the individual names. It also performs the reverse function.

Use the

System.Web.UI.WebControls.FontNamesConverter.ConvertFrom(System.Co mponentModel.ITypeDescriptorContext,System.Globalization.CultureInfo,S ystem.Object) method of this class to convert a single string containing a list of font names to an array of strings containing the individual names. Each font name in the string must be separated by a comma. For example, the string, "arial, times new roman, verdana", converts to an array that contains the strings "arial", "times new roman", and "verdana". Notice the commas are removed along with any white space at the beginning or end of the font name. White space in the middle of a font name is not removed.

**FontNamesConverter** 

Example Syntax:

**ToString** 

[C#] public FontNamesConverter();

[C++] public: FontNamesConverter();

[VB] Public Sub New()

[JScript] public function FontNamesConverter();

CanConvertFrom

[C#] public override bool CanConvertFrom(ITypeDescriptorContext context,

Type sourceType);

[C++] public: bool CanConvertFrom(ITypeDescriptorContext\* context, Type\*

21

22

23

24

sourceType);

[VB] Overrides Public Function CanConvertFrom(ByVal context As

ITypeDescriptorContext, ByVal sourceType As Type) As Boolean

[JScript] public override function CanConvertFrom(context:

ITypeDescriptorContext, sourceType: Type): Boolean;

Description

Determines whether this converter can convert an object of the specified data type to an array of strings containing individual font names.

Return Value: true if the type can be converted; otherwise, false.

Use the

System.Web.UI.WebControls.FontNamesConverter.CanConvertFrom(Syste m.ComponentModel.ITypeDescriptorContext,System.Type) method to determine whether the specified data type can be converted to an array of strings containing individual font names. A

System.ComponentModel.ITypeDescriptorContext that provides information about the context of a type converter. You can optionally pass in null for this parameter. A **System.Type** that represents the data type to convert from. This parameter must be set to the string data type.

ConvertFrom

[C#] public override object ConvertFrom(ITypeDescriptorContext context,

CultureInfo culture, object value);

[C++] public: Object\* ConvertFrom(ITypeDescriptorContext\* context,

CultureInfo\* culture, Object\* value);

[VB] Overrides Public Function ConvertFrom(ByVal context As

ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)

As Object

[JScript] public override function ConvertFrom(context : ITypeDescriptorContext, culture : CultureInfo, value : Object) : Object;

## Description

Converts a string that represents a list of font names into an array of strings containing individual font names.

Return Value: A System.Object that represents the array of strings containing the individual font names.

Use the

System.Web.UI.WebControls.FontNamesConverter.ConvertFrom(System.ComponentModel.ITypeDescriptorContext,System.Globalization.CultureInfo,System.Object) method to convert a single string containing a list of font names to an array of strings containing the individual names. Each font name in the string must be separated by a comma. For example, the string, "arial, times new roman, verdana", converts to an array that contains the strings "arial", "times new roman", and "verdana". Notice that the commas are removed along with any white space at the beginning or end of the font name. White space in the middle of a font name is not removed. A System.ComponentModel.ITypeDescriptorContext that provides information about the context of a type converter. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in null for this parameter. A System.Globalization.CultureInfo object that represents information about a culture such as language, calendar

system, and so on. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in **null** for this parameter.

A **System.Object** that represents the source string to convert from.

ConvertTo

[C#] public override object ConvertTo(ITypeDescriptorContext context,

CultureInfo culture, object value, Type destinationType);

[C++] public: Object\* ConvertTo(ITypeDescriptorContext\* context, CultureInfo\* culture, Object\* value, Type\* destinationType);

[VB] Overrides Public Function ConvertTo(ByVal context As

ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object,

ByVal destinationType As Type) As Object

[JScript] public override function ConvertTo(context : ITypeDescriptorContext, culture : CultureInfo, value : Object, destinationType : Type) : Object;

Description

Creates a string that represents a list of font names from an array of strings containing individual font names.

Return Value: A System.Object that represents a string containing a list of font names.

Use the

System.Web.UI.WebControls.FontNamesConverter.ConvertTo(System.ComponentModel.ITypeDescriptorContext,System.Globalization.CultureInfo,System.Object,System.Type) method to convert an array of strings containing the individual font names to a single string containing a list of the names. For

"verdana" converts to the string, "arial, times new roman, and "verdana" converts to the string, "arial, times new roman, verdana". Notice that commas are automatically inserted between the font names without any white space. A System.ComponentModel.ITypeDescriptorContext object that provides information about the context of a type converter. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in null for this parameter. A System.Globalization.CultureInfo object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in null for this parameter. A System.Object that represents the source array of strings to convert from. A System.Type object that represents the data type to convert to. This parameter must be set to the string data type.

FontSize enumeration (System.Web.UI.WebControls)
ToString

## Description

Specifies the font sizes defined by HTML 4.0.

The **System.Web.UI.WebControls.FontSize** enumeration represents the font sizes defined by HTML 4.0. The font size can be an absolute or relative size.

**ToString** 

[C#] public const FontSize AsUnit;

[C++] public: const FontSize AsUnit;

	1	[VB] Public Const AsUnit As FontSize
	2	[JScript] public var AsUnit : FontSize;
	3	
	4	Description
	5	The font size is specified by a point value.
	6	ToString
	7	
	8	[C#] public const FontSize Large;
	9	[C++] public: const FontSize Large;
	10	[VB] Public Const Large As FontSize
And the first from the first first that	11	[JScript] public var Large : FontSize;
	12	
	13	Description
	14	The font size is two sizes larger than the base font size.
	15	ToString
	16	
diz	17	[C#] public const FontSize Larger;
	18	[C++] public: const FontSize Larger;
	19	[VB] Public Const Larger As FontSize
	20	[JScript] public var Larger : FontSize;
	21	
	22	Description
	23	The font size is one size larger than the parent element.
	24	ToString

П	
1	
2	[C#] public const FontSize Medium;
3	[C++] public: const FontSize Medium;
4	[VB] Public Const Medium As FontSize
5	[JScript] public var Medium : FontSize;
6	
7	Description
8	The font size is one size larger than the default font size.
9	ToString
10	
11	[C#] public const FontSize NotSet;
12	[C++] public: const FontSize NotSet;
13	[VB] Public Const NotSet As FontSize
14	[JScript] public var NotSet : FontSize;
15	
16	Description
17	The font size is not set.
18	ToString
19	
20	[C#] public const FontSize Small;
21	[C++] public: const FontSize Small;
22	[VB] Public Const Small As FontSize
23	[JScript] public var Small : FontSize;
24	
25	Description

1	The base font size determined by the browser.
2	ToString
3	
4	[C#] public const FontSize Smaller;
5	[C++] public: const FontSize Smaller;
6	[VB] Public Const Smaller As FontSize
7	[JScript] public var Smaller : FontSize;
8	
9	Description
10	The font size is one size smaller than the parent element.
11	ToString
12	
13	[C#] public const FontSize XLarge;
14	[C++] public: const FontSize XLarge;
15	[VB] Public Const XLarge As FontSize
16	[JScript] public var XLarge : FontSize;
17	
18	Description
19	The font size is three sizes larger than the base font size.
20	ToString
21	
22	[C#] public const FontSize XSmall;
23	[C++] public: const FontSize XSmall;
24	[VB] Public Const XSmall As FontSize
25	[JScript] public var XSmall : FontSize;

	1	
	2	Description
	3	The font size is one size smaller than the base font size.
	4	ToString
	5	
	6	[C#] public const FontSize XXLarge;
	7	[C++] public: const FontSize XXLarge;
	8	[VB] Public Const XXLarge As FontSize
	9	[JScript] public var XXLarge : FontSize;
	10	
j	11	Description
	12	The font size is four sizes larger than the base font size.
	13	ToString
H	14	
	15	[C#] public const FontSize XXSmall;
	16	[C++] public: const FontSize XXSmall;
	17	[VB] Public Const XXSmall As FontSize
	18	[JScript] public var XXSmall : FontSize;
	19	
	20	Description
	21	The font size is two sizes smaller than the base font size.
	22	FontUnit structure (System.Web.UI.WebControls)
	23	ToString
	24	

19 20

22

21

23

24

25

Description

Represents the size of a font.

For a list of initial property values for an instance of

System.Web.UI.WebControls.FontUnit, see the

System.Web.UI.WebControls.FontUnit.#ctor constructor.

**ToString** 

[C#] public static readonly FontUnit Empty;

[C++] public: static FontUnit Empty;

[VB] Public Shared ReadOnly Empty As FontUnit

[JScript] public static var Empty : FontUnit;

Description

Represents an empty System. Web.UI. WebControls. FontUnit object. This field is read only.

Use the System. Web.UI. WebControls. FontUnit. Empty field to represent an empty System. Web. UI. Web Controls. Font Unit object. No HTML rendered output is created when this field is used.

**ToString** 

[C#] public static readonly FontUnit Large;

[C++] public: static FontUnit Large;

[VB] Public Shared ReadOnly Large As FontUnit

	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
j	10	
	11	
	12	l
	13	
	14	
-  -  -	15	
	16	
	17	
	18	
	19	
	20	
	21	

[JScript] public static var Large : FontUnit; Description Represents a System. Web. UI. WebControls. Font Unit object with the System.Web.UI.WebControls.FontUnit.Type property set to FontSize.Large font. This field is read only. Use the System. Web. UI. Web Controls. Font Unit. Large field to represent a System. Web. UI. Web Controls. Font Unit object with the  ${\bf System. Web. UI. WebControls. Font Unit. Type\ property\ set\ to\ Font Size. Large\ .}$ **ToString** [C#] public static readonly FontUnit Larger; [C++] public: static FontUnit Larger; [VB] Public Shared ReadOnly Larger As FontUnit [JScript] public static var Larger: FontUnit; Description Represents a System. Web.UI. WebControls. Font Unit object with the  ${\bf System. Web. UI. WebControls. Font Unit. Type\ property\ set\ to\ Font Size. Larger}$ font. This field is read only. Use the System. Web.UI. WebControls. FontUnit. Larger field to represent a System. Web. UI. Web Controls. Font Unit object with the 22 System.Web.UI.WebControls.FontUnit.Type property set to FontSize.Larger . 23

**ToString** 

24

25

18

20

21

22

23 24

25

[C#] public static readonly FontUnit Medium;

[C++] public: static FontUnit Medium;

[VB] Public Shared ReadOnly Medium As FontUnit

[JScript] public static var Medium : FontUnit;

Description

Represents a System.Web.UI.WebControls.FontUnit object with the System.Web.UI.WebControls.FontUnit.Type property set to FontSize.Medium font. This field is read only.

Use the  ${\bf System.Web.UI.WebControls.FontUnit.Medium}$  field to represent a System. Web.UI. WebControls. FontUnit object with the System.Web.UI.WebControls.FontUnit.Type property set to FontSize.Medium

**ToString** 

[C#] public static readonly FontUnit Small;

[C++] public: static FontUnit Small;

[VB] Public Shared ReadOnly Small As FontUnit

[JScript] public static var Small : FontUnit;

Description

Represents a System. Web.UI. WebControls. FontUnit object with the  ${\bf System. Web. UI. WebControls. Font Unit. Type\ property\ set\ to\ Font Size. Small}$ font. This field is read only.

	1	
	2	a
	3	S
	4	
	5	
	6	[
	7	
	8	[
	9	
	10	
	11	ا ا
	12	
	13	
	14	
ايرا '	15	
	16	
i da	17	
	18	
	19	

23

24

25

Use the System.Web.UI.WebControls.FontUnit.Small field to represent a System.Web.UI.WebControls.FontUnit object with the System.Web.UI.WebControls.FontUnit.Type property set to FontSize.Small. **ToString** 

[C#] public static readonly FontUnit Smaller;

[C++] public: static FontUnit Smaller;

[VB] Public Shared ReadOnly Smaller As FontUnit

[JScript] public static var Smaller: FontUnit;

Description

Represents a System. Web. UI. Web Controls. Font Unit object with the  ${\bf System. Web. UI. Web Controls. Font Unit. Type\ property\ set\ to\ Font Size. Smaller}$ font. This field is read only.

Use the System.Web.UI.WebControls.FontUnit.Smaller field to represent a System. Web.UI. WebControls. FontUnit object with the System.Web.UI.WebControls.FontUnit.Type property set to FontSize.Smaller

**ToString** 

[C#] public static readonly FontUnit XLarge;

[C++] public: static FontUnit XLarge;

[VB] Public Shared ReadOnly XLarge As FontUnit

[JScript] public static var XLarge : FontUnit;

3

8

19

17

21

23

\_

## Description

Represents a **System.Web.UI.WebControls.FontUnit** object with the **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.XLarge** font. This field is read only.

Use the System.Web.UI.WebControls.FontUnit.XLarge field to represent a System.Web.UI.WebControls.FontUnit object with the System.Web.UI.WebControls.FontUnit.Type property set to FontSize.XLarge.

**ToString** 

[C#] public static readonly FontUnit XSmall;

[C++] public: static FontUnit XSmall;

[VB] Public Shared ReadOnly XSmall As FontUnit

[JScript] public static var XSmall : FontUnit;

## Description

Represents a **System.Web.UI.WebControls.FontUnit** object with the **System.Web.UI.WebControls.FontUnit.Type** property set to **FontSize.XSmall** font. This field is read only.

Use the System.Web.UI.WebControls.FontUnit.XSmall field to represent a System.Web.UI.WebControls.FontUnit object with the System.Web.UI.WebControls.FontUnit.Type property set to FontSize.XSmall.

**ToString** 

[C#] public static readonly FontUnit XXLarge;

1	[C++] public: static FontUnit XXLarge;
2	[VB] Public Shared ReadOnly XXLarge As FontUnit
3	[JScript] public static var XXLarge : FontUnit;
4	
5	Description
6	Represents a System.Web.UI.WebControls.FontUnit object with the
7	System.Web.UI.WebControls.FontUnit.Type property set to
8	FontSize.XXLarge font. This field is read only.
9	Use the System.Web.UI.WebControls.FontUnit.XXLarge field to
10	represent a System.Web.UI.WebControls.FontUnit object with the
11	System.Web.UI.WebControls.FontUnit.Type property set to
12	FontSize.XXLarge .
13	ToString
14	
. 15	[C#] public static readonly FontUnit XXSmall;
16	[C++] public: static FontUnit XXSmall;
17	[VB] Public Shared ReadOnly XXSmall As FontUnit
18	[JScript] public static var XXSmall : FontUnit;
19	
20	Description
21	Represents a System.Web.UI.WebControls.FontUnit object with the
22	System.Web.UI.WebControls.FontUnit.Type property set to
23	FontSize.XXSmall font. This field is read only.
24	Use the System.Web.UI.WebControls.FontUnit.XXSmall field to

represent a System.Web.UI.WebControls.FontUnit object with the

20

21

22

23

24

25

[VB] Public Sub New(ByVal value As Integer)

[JScript] public function FontUnit(value : int);

Description

Initializes a new instance of the **System.Web.UI.WebControls.FontUnit** class with the specified font size.

Use this constructor to create and initialize an instance of the **System.Web.UI.WebControls.FontUnit** class using the specified font size. An integer that represents the size of the font.

**FontUnit** 

Example Syntax:

**ToString** 

[C#] public FontUnit(string value);

[C++] public: FontUnit(String\* value);

[VB] Public Sub New(ByVal value As String)

[JScript] public function FontUnit(value : String);

Description

Initializes a new instance of the **System.Web.UI.WebControls.FontUnit** class with the specified font value.

Use this constructor to create and initialize an instance of the 
System.Web.UI.WebControls.FontUnit class using the specified font value. A 
string representation of one of the System.Web.UI.WebControls.FontSize value.

**FontUnit** 

1	Example Syntax:
2	ToString
3	
4	[C#] public FontUnit(Unit value);
5	[C++] public: FontUnit(Unit value);
6	[VB] Public Sub New(ByVal value As Unit)
7	[JScript] public function FontUnit(value : Unit);
8	
9	Description
10	Initializes a new instance of the System.Web.UI.WebControls.FontUnit
10 11 12 13	class with the specified System.Web.UI.WebControls.Unit object.
[U] (II) 12	Use this constructor to create and initialize an instance of the
13 13	System.Web.UI.WebControls.FontUnit class using the specified
14	System.Web.UI.WebControls.Unit object. A
14 14 15	System.Web.UI.WebControls.Unit object that specifies the font size.
[] [] 16	FontUnit
ļ š 17	Example Syntax:
18	ToString
19	
20	[C#] public FontUnit(string value, CultureInfo culture);
21	[C++] public: FontUnit(String* value, CultureInfo* culture);
22	[VB] Public Sub New(ByVal value As String, ByVal culture As CultureInfo)
23	[JScript] public function FontUnit(value : String, culture : CultureInfo);
24	
25	Description

1	[VB] Public ReadOnly Property Type As FontSize
2	[JScript] public function get Type() : FontSize;
3	
4	Description
5	Gets a System.Web.UI.WebControls.FontSize object that represents the
6	font size.
7	Use the System.Web.UI.WebControls.FontUnit.Type property to
8	determine the font size.
9	Unit
10	ToString
10 11 12	
12	[C#] public Unit Unit {get;}
13 13	[C++] public:property Unit get_Unit();
14	[VB] Public ReadOnly Property Unit As Unit
≔15	[JScript] public function get Unit(): Unit;
] []16	
17	Description
18	Gets a System.Web.UI.WebControls.Unit object that represents the font
19	size.
20	Equals
21	
22	[C#] public override bool Equals(object obj);
23	[C++] public: bool Equals(Object* obj);
24	[VB] Overrides Public Function Equals(ByVal obj As Object) As Boolean
25	[JScript] public override function Equals(obj : Object) : Boolean;

Description

Determines whether the specified **System.Object** is equivilent to this instance of the **System.Web.UI.WebControls.FontUnit** class.

Return Value: true if the specified System.Object is equivilent to this instance of the System.Web.UI.WebControls.FontUnit class; otherwise, false. A System.Object that contains the object to compare to this instance.

GetHashCode

```
[C#] public override int GetHashCode();
```

```
[C++] public: int GetHashCode();
```

[VB] Overrides Public Function GetHashCode() As Integer

[JScript] public override function GetHashCode(): int;

Description

Returns the hash code for this instance.

Return Value: A 32-bit signed integer hash code.

op\_Equality

[C#] public static bool operator ==(FontUnit left, FontUnit right);

[C++] public: static bool op\_Equality(FontUnit left, FontUnit right);

[VB] returnValue = FontUnit.op Equality(left, right)

[JScript] returnValue = left == right;

Description

24

Compares two System. Web. UI. Web Controls. Font Unit objects for equality. Return Value: true if both System. Web. UI. WebControls. Font Unit objects are equal; otherwise, false . A System. Web. UI. Web Controls. Font Unit object that contains font properties on the left of the operator. A System.Web.UI.WebControls.FontUnit object that contains font properties on the right of the operator. op Implicit [C#] public static implicit operator FontUnit(int n); [C++] public: static FontUnit op Implicit(int n); [VB] returnValue = FontUnit.op Implicit(n) [JScript] returnValue = n; Description Implicitly creates a System.Web.UI.WebControls.FontUnit of type System.Drawing.Point from an integer value. An integer representing the System.Drawing.Point value to convert into a System.Web.UI.WebControls.FontUnit. op Inequality [C#] public static bool operator !=(FontUnit left, FontUnit right); [C++] public: static bool op\_Inequality(FontUnit left, FontUnit right);

[VB] returnValue = FontUnit.op\_Inequality(left, right)

[JScript] returnValue = left != right;

1	
2	Descriptio
3	Con
4	inequality.
5	Return Val
6	not equal;
7	contains fo
8	System.W
9	of the oper
10	Pars
11	
[] [] <sup>12</sup>	[C#] public
[] []13	[C++] pub
" []]14	[VB] Publi
`*	[JScript] p
16	string to its
17	
18	Descriptio
19	Con
20	System.W
21	one of the
22	Pars

n

mpares two System. Web. UI. Web Controls. Font Unit objects for

lue: true if both System. Web. UI. Web Controls. Font Unit objects are otherwise, false . A System. Web. UI. Web Controls. Font Unit that ont properties on the left of the operator. A

eb.UI.WebControls.FontUnit that contains font properties on the right rator.

se

c static FontUnit Parse(string s);

lic: static FontUnit Parse(String\* s);

ic Shared Function Parse(ByVal s As String) As FontUnit public static function Parse(s : String) : FontUnit; Converts the specified s System.Web.UI.WebControls.FontUnit equivalent.

n

enverts the specified string to its default

eb.UI.WebControls.FontUnit equivalent. A string representation of System.Web.UI.WebControls.FontSize value.

se

[C#] public static FontUnit Parse(string s, CultureInfo culture);

[C++] public: static FontUnit Parse(String\* s, CultureInfo\* culture);

21 22

23

25

24

[VB] Public Shared Function Parse(ByVal s As String, ByVal culture As CultureInfo) As FontUnit

[JScript] public static function Parse(s: String, culture: CultureInfo): FontUnit;

Description

Converts the specified string to its

System.Web.UI.WebControls.FontUnit equivalent in the specified culture. A string representation of one of the System. Web.UI. WebControls. Font Size value.

A System. Globalization. Culture Info object that represents the culture of the System.Web.UI.WebControls.FontUnit object.

Point

[C#] public static FontUnit Point(int n);

[C++] public: static FontUnit Point(int n);

[VB] Public Shared Function Point(ByVal n As Integer) As FontUnit

[JScript] public static function Point(n:int): FontUnit;

Description

Creates a System. Web. UI. Web Controls. Font Unit of type System.Drawing.Point from an integer value.

Return Value: A System. Web. UI. WebControls. Font Unit that represents the new font size. An integer representing the System.Drawing.Point value to convert into a System.Web.UI.WebControls.FontUnit.

**ToString** 

1

2

[C#] public override string ToString();
[C++] public: String\* ToString();
[VB] Overrides Public Function ToString() As String
[JScript] public override function ToString(): String; Converts the
System.Web.UI.WebControls.FontUnit object to a string representation.

Description

Converts the **System.Web.UI.WebControls.FontUnit** object to the default string representation.

Return Value: A string representation of the

System. Web. UI. Web Controls. Font Unit object.

**ToString** 

[C#] public string ToString(CultureInfo culture);

[C++] public: String\* ToString(CultureInfo\* culture);

[VB] Public Function ToString(ByVal culture As CultureInfo) As String

[JScript] public function ToString(culture : CultureInfo) : String;

Description

Converts the System.Web.UI.WebControls.FontUnit object to a string representation using the specified System.Globalization.CultureInfo object. A System.Globalization.CultureInfo object that contains the culture.

FontUnitConverter class (System.Web.UI.WebControls)

**ToString** 

21

22

23

24

Determines if the specified data type can be converted to a System.Web.UI.WebControls.FontUnit.

Return Value: true if the type can be converted; otherwise false. An System.ComponentModel.ITypeDescriptorContext that provides information about the context of a type converter. A System.Type that represents the data type to check.

ConvertFrom

[C#] public override object ConvertFrom(ITypeDescriptorContext context, CultureInfo culture, object value);

[C++] public: Object\* ConvertFrom(ITypeDescriptorContext\* context,

CultureInfo\* culture, Object\* value);

[VB] Overrides Public Function ConvertFrom(ByVal context As

ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)

As Object

[JScript] public override function ConvertFrom(context : ITypeDescriptorContext, culture : CultureInfo, value : Object) : Object;

Description

Converts the specified System.Object into a

System.Web.UI.WebControls.FontUnit . An

**System.ComponentModel.ITypeDescriptorContext** that provides information about the context of a type converter. A **System.Globalization.CultureInfo** object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for

future versions of this method. You can optionally pass in **null** for this parameter. The **System.Object** to convert to a **System.Web.UI.WebControls.FontUnit**.

ConvertTo

[C#] public override object ConvertTo(ITypeDescriptorContext context, CultureInfo culture, object value, Type destinationType);

[C++] public: Object\* ConvertTo(ITypeDescriptorContext\* context, CultureInfo\* culture, Object\* value, Type\* destinationType);

[VB] Overrides Public Function ConvertTo(ByVal context As

ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object, ByVal destinationType As Type) As Object

[JScript] public override function ConvertTo(context : ITypeDescriptorContext, culture : CultureInfo, value : Object, destinationType : Type) : Object;

#### Description

Converts the specified **System.Web.UI.WebControls.FontUnit** into the specified **System.Type** . An

System.ComponentModel.ITypeDescriptorContext that provides information about the context of a type converter. A System.Globalization.CultureInfo object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in null for this parameter.

A System.Object that represents the source array of strings to convert from. A System.Type that represents the data type to convert to.

**GetStandardValues** 

1	
2	[C#] public override StandardValues0
3	GetStandardValues(ITypeDescriptor(
4	[C++] public: StandardValuesCollect
5	GetStandardValues(ITypeDescriptor(
6	[VB] Overrides Public Function GetS
7	ITypeDescriptorContext) As Standard
8	[JScript] public override function Get
9	ITypeDescriptorContext): StandardV
10	
11	Description
42 13	Returns a
3	System.ComponentModel.TypeCor
]4	containing standard System.Web.UI.
]4 -1	containing standard System.Web.UI.  Return Value: A
14 15 16	
14 45	Return Value: A
14 15 16	Return Value: A  System.ComponentModel.TypeCor
14 15 6	Return Value: A  System.ComponentModel.TypeCor  containing System.Web.UI.WebCor
14 15 6 17	Return Value: A  System.ComponentModel.TypeComponenting System.Web.UI.WebComponentModel.ITypeDe
17 18	Return Value: A  System.ComponentModel.TypeComponenting System.Web.UI.WebComponentModel.ITypeDe  about the context of a type converter.
14 15 16 17 18	Return Value: A  System.ComponentModel.TypeComponenting System.Web.UI.WebComponentModel.ITypeDe  about the context of a type converter.
14 15 17 18 19 20	Return Value: A  System.ComponentModel.TypeComponenting System.Web.UI.WebComponentModel.ITypeDemodel.T
17 18 19 20 21	Return Value: A  System.ComponentModel.TypeComponenting System.Web.UI.WebComponentModel.ITypeDemodel.Allower about the context of a type converter.  GetStandardValuesExclusive  [C#] public override bool GetStandard

[C#] public override StandardValuesCollection
GetStandardValues(ITypeDescriptorContext context);
[C++] public: StandardValuesCollection*
GetStandardValues(ITypeDescriptorContext* context);
[VB] Overrides Public Function GetStandardValues(ByVal context As
ITypeDescriptorContext) As StandardValuesCollection
[JScript] public override function GetStandardValues(context:
ITypeDescriptorContext): StandardValuesCollection;
Description
Returns a
System. Component Model. Type Converter. Standard Values Collection
containing standard System.Web.UI.WebControls.FontUnit values.
Return Value: A
System. Component Model. Type Converter. Standard Values Collection
containing System.Web.UI.WebControls.FontUnit values. An
System.ComponentModel.ITypeDescriptorContext that provides information

public override bool GetStandardValuesExclusive(ITypeDescriptorContext ext);

+] public: bool GetStandardValuesExclusive(ITypeDescriptorContext\* 25 context);

1	[VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As
2	ITypeDescriptorContext) As Boolean
3	[JScript] public override function GetStandardValuesExclusive(context:
4	ITypeDescriptorContext): Boolean;
5	
6	Description
7	Indicates whether the specified context contains exclusive standard values.
8	Return Value: false for all cases. An
9	System.ComponentModel.ITypeDescriptorContext that provides information
10	about the context of a type converter.
	GetStandardValuesSupported
]3	[C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext
]4	context);
14   15   16	[C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext*
16	context);
17	[VB] Overrides Public Function GetStandardValuesSupported(ByVal context As
18	ITypeDescriptorContext) As Boolean
19	[JScript] public override function GetStandardValuesSupported(context:
20	ITypeDescriptorContext): Boolean;
21	
22	Description
23	Indicates whether the specified context contains suppurted standard values.
24	Return Value: true for all cases. An
25	

1	System.ComponentModel.ITypeDescriptorContext that provides information
2	about the context of a type converter.
3	GridLines enumeration (System.Web.UI.WebControls)
4	ToString
5	
6	
7	Description
8	Specifies the grid line styles for controls displaying items in a table
9	structure.
Ð	The System.Web.UI.WebControls.GridLines enumeration represents the
	grid line styles for controls displaying items in a table structure.
	ToString
13	
#4	[C#] public const GridLines Both;
14 15 16	[C++] public: const GridLines Both;
16	[VB] Public Const Both As GridLines
17	[JScript] public var Both : GridLines;
18	
19	Description
20	Both horizontal and vertical grid lines rendered.
21	ToString
22	
23	[C#] public const GridLines Horizontal;
24	[C++] public: const GridLines Horizontal;
25	[VB] Public Const Horizontal As GridLines

1	[JScript] public var Horizontal : GridLines;
2	
3	Description
4	Only horizontal grid lines rendered.
5	ToString
6	
7	[C#] public const GridLines None;
8	[C++] public: const GridLines None;
9	[VB] Public Const None As GridLines
10	[JScript] public var None : GridLines;
10 11 12 13	
12	Description
<b>1</b> 3	No grid lines rendered.
1	ToString
14	
16	[C#] public const GridLines Vertical;
17	[C++] public: const GridLines Vertical;
18	[VB] Public Const Vertical As GridLines
19	[JScript] public var Vertical : GridLines;
20	
21	Description
22	Only vertical grid lines rendered.
23	HorizontalAlign enumeration (System.Web.UI.WebControls)
24	ToString
25	

21

22

23

24

25

Description

Specifies the horizonal alignment of items within a container.

The **System.Web.UI.WebControls.HorizontalAlign** enumeration represents the horizontal alignment options for items within a container, such as a **System.Web.UI.WebControls.TableCell**.

**ToString** 

[C#] public const HorizontalAlign Center;

[C++] public: const HorizontalAlign Center;

[VB] Public Const Center As HorizontalAlign

[JScript] public var Center : HorizontalAlign;

Description

The contents of a container are centered.

**ToString** 

[C#] public const HorizontalAlign Justify;

[C++] public: const HorizontalAlign Justify;

[VB] Public Const Justify As HorizontalAlign

[JScript] public var Justify : HorizontalAlign;

Description

The contents of a container are uniformly spread out and aligned with both the left and right margins.

**ToString** 

[C#] public const HorizontalAlign Left;

[C++] public: const HorizontalAlign Left;

[VB] Public Const Left As HorizontalAlign

[JScript] public var Left : HorizontalAlign;

## Description

The contents of a container are left justified.

**ToString** 

[C#] public const HorizontalAlign NotSet;

[C++] public: const HorizontalAlign NotSet;

[VB] Public Const NotSet As HorizontalAlign

[JScript] public var NotSet : HorizontalAlign;

# Description

The horizontal alignment is not set.

**ToString** 

[C#] public const HorizontalAlign Right;

[C++] public: const HorizontalAlign Right;

[VB] Public Const Right As HorizontalAlign

```
[JScript] public var Right: HorizontalAlign;
 2
     Description
 3
           The contents of a container are right justified.
 4
           HyperLink class (System.Web.UI.WebControls)
 5
            ToString
 6
 7
 8
     Description
 9
            A control that displays a link to another Web page.
10
11
12
13
            Use the System.Web.UI.WebControls.HyperLink control to create a link
     to another Web page. The System.Web.UI.WebControls.HyperLink control is
     typically displayed as text specified by the
     System.Web.UI.WebControls.HyperLink.Text property. It can also be
     displayed as an image specified by the
      {\bf System. Web. UI. WebControls. HyperLink. Image Url\ property.}
            HyperLink
 17
            Example Syntax:
 18
             ToString
 19
 20
      [C#] public HyperLink();
 21
      [C++] public: HyperLink();
  22
      [VB] Public Sub New()
  23
      [JScript] public function HyperLink();
  24
```

Description

Initializes a new instance of the

System.Web.UI.WebControls.HyperLink class.

Use this constructor to create and initialize a new instance of the

System.Web.UI.WebControls.HyperLink control.

AccessKey

Attributes

BackColor

BorderColor

BorderStyle

BorderWidth

ChildControlsCreated

ClientID

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

19

20

21

22

23

24

Enabled

EnableViewState

**Events** 

Font

ForeColor

HasChildViewState

Height

ID

ImageUrl

**ToString** 

#### Description

Gets or sets the path to an image to display for the

System. Web. UI. Web Controls. HyperLink control.

The **System.Web.UI.WebControls.HyperLink** control can be displayed as text or an image. Use the

System.Web.UI.WebControls.HyperLink.ImageUrl property to specify an image to display for the System.Web.UI.WebControls.HyperLink control.

IsTrackingViewState

NamingContainer

NavigateUrl

**ToString** 

### Description

Gets or sets the URL to link to when the

System.Web.UI.WebControls.HyperLink control is clicked.

Use the **System.Web.UI.WebControls.HyperLink.NavigateUrl** property to specify the URL to navigate to when the

 ${\bf System. Web. UI. WebControls. HyperLink}\ \ {\bf control}\ \ is\ {\bf clicked}.$ 

19

20

21

22

23

24

25

Page

Parent

Site

Style

TabIndex

TagKey

TagName

Target

**ToString** 

## Description

Gets or sets the target window or frame to display the Web page content linked to when the **System.Web.UI.WebControls.HyperLink** control is clicked.

Use the **System.Web.UI.WebControls.HyperLink.Target** property to specify the frame or window that displays the Web page linked to when the **System.Web.UI.WebControls.HyperLink** control is clicked. The Web page is specified by setting the **System.Web.UI.WebControls.HyperLink.NavigateUrl** property.

TemplateSourceDirectory

Text

**ToString** 

Description

Gets or sets the text caption for the

System.Web.UI.WebControls.HyperLink control.

The **System.Web.UI.WebControls.HyperLink** control can be displayed as text or an image. Use the **System.Web.UI.WebControls.HyperLink.Text** property to specify the text to display for the

System.Web.UI.WebControls.HyperLink control.

ToolTip

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

[C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As

HtmlTextWriter)

[JScript] protected override function AddAttributesToRender(writer:

HtmlTextWriter);

Description

Adds the attribututes of the a **System.Web.UI.WebControls.HyperLink** to the output stream for rendering. The output stream to render on the client.

AddParsedSubObject

[C#] protected override void AddParsedSubObject(object obj);
[C++] protected: void AddParsedSubObject(Object* obj);
[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)
[JScript] protected override function AddParsedSubObject(obj : Object);
LoadViewState
[C#] protected override void LoadViewState(object savedState);
[C++] protected: void LoadViewState(Object* savedState);
[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)
[JScript] protected override function LoadViewState(savedState : Object);
Description
Load previously saved state. Overridden to synchronize Text property with
LiteralContent.
RenderContents
[C#] protected override void RenderContents(HtmlTextWriter writer);
[C++] protected: void RenderContents(HtmlTextWriter* writer);
[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)
[JScript] protected override function RenderContents(writer: HtmlTextWriter);
Description
Displays the <b>System.Web.UI.WebControls.HyperLink</b> on a page. The
output stream to render on the client.

	1	
	2	
	3	
	4	
	5	
	6	
	7	
	8	
	9	
i king	0	
	1	
	2	
اچا اور	3	
#  []1  ]	4	
	5	
	6	
   	7	
1	8	
1	9	
2	20	
		11

HyperLinkColumn class (System.Web.UI.WebControls)

**TrackViewState** 

Description

A column type for the System.Web.UI.WebControls.DataGrid control

that contains a hyperlink for each item in the column.

Use the System.Web.UI.WebControls.HyperLinkColumn column type in a System.Web.UI.WebControls.DataGrid control to create a hyperlink for each row in the System.Web.UI.WebControls.DataGrid control. Set the System.Web.UI.WebControls.HyperLinkColumn.Text property to specify the caption text for the hyperlink. To specify the URL to navigate to when the hyperlink is clicked, set the

System.Web.UI.WebControls.HyperLinkColumn.NavigateUrl property.

HyperLinkColumn

Example Syntax:

**TrackViewState** 

[C#] public HyperLinkColumn();

[C++] public: HyperLinkColumn();

[VB] Public Sub New()

[JScript] public function HyperLinkColumn();

Description

25

22

23

24

Initializes a new instance of the 1 System.Web.UI.WebControls.HyperLinkColumn class. 2 Use this constructor to create and initialize a new instance of the 3 System.Web.UI.WebControls.HyperLinkColumn class. 4 DataNavigateUrlField 5 **TrackViewState** 6 7 [C#] public virtual string DataNavigateUrlField {get; set;} 8 [C++] public: \_\_property virtual String\* get\_DataNavigateUrlField();public: 9 property virtual void set DataNavigateUrlField(String\*); 10 [VB] Overridable Public Property DataNavigateUrlField As String 11 [JScript] public function get DataNavigateUrlField(): String;public function set 12 DataNavigateUrlField(String); 13 14 Description 15 Gets or sets the field name from a data source to bind to the URL of the 16 hyperlinks in System. Web. UI. Web Controls. HyperLink Column column. 17 DataNavigateUrlFormatString 18 **TrackViewState** 19 20 [C#] public virtual string DataNavigateUrlFormatString {get; set;} 21 [C++] public: property virtual String\* 22 get\_DataNavigateUrlFormatString();public: \_\_property virtual void 23 set DataNavigateUrlFormatString(String\*); 24 [VB] Overridable Public Property DataNavigateUrlFormatString As String

	1	[JScript] public function get DataNavigateUrlFormatString(): String;public
	2	function set DataNavigateUrlFormatString(String);
	3	
	4	Description
	5	Gets or sets the string that specifies the display format for the URL of the
	6	hyperlinks in the System.Web.UI.WebControls.HyperLinkColumn column.
	7	Use the
	8	System.Web.UI.WebControls.HyperLinkColumn.DataNavigateUrlFormatSt
	9	ing property to provide a custom format for the URL of the hyperlinks in the
	10	System.Web.UI.WebControls.HyperLinkColumn column.
u	11	DataTextField
	12	TrackViewState
	13	
	14	[C#] public virtual string DataTextField {get; set;}
	15	[C++] public:property virtual String* get_DataTextField();public:property
	16	virtual void set_DataTextField(String*);
•	17	[VB] Overridable Public Property DataTextField As String
	18	[JScript] public function get DataTextField(): String;public function set
	19	DataTextField(String);
	20	
	21	Description
	22	Gets or sets the field name from a data source to bind to the text caption of
	23	the hyperlinks in System. Web. UI. Web Controls. HyperLink Column column.
	24	DataTextFormatString
	25	TrackViewState

[C#] public virtual string DataTextFormatString {get; set;} [C++] public: property virtual String\* get DataTextFormatString();public: property virtual void set DataTextFormatString(String\*); [VB] Overridable Public Property DataTextFormatString As String 5 [JScript] public function get DataTextFormatString(): String; public function set 6 DataTextFormatString(String); 7 8 Description 9 Gets or sets the string that specifies the display format for the text caption 10 of the hyperlinks in the System. Web. UI. WebControls. HyperLinkColumn 11 column. 12 Use the 13 System.Web.UI.WebControls.HyperLinkColumn.DataTextFormatString 14 property to provide a custom format for the items in the column. 15 DesignMode 16 FooterStyle 17 FooterText 18 HeaderImageUrl 19 HeaderStyle 20 HeaderText 21 IsTrackingViewState 22 ItemStyle 23 NavigateUrl 24

25

TrackViewState

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Description

2

3

5

Gets or sets the URL to link to when a hyperlink in the column is clicked.

Use the **System.Web.UI.WebControls.HyperLinkColumn.NavigateUrl** property to specify the URL to navigate to when a hyperlink in the column is clicked.

Owner

SortExpression

Target

**TrackViewState** 

Description

Gets or sets the target window or frame to display the Web page content linked to when the hyperlink in the column is clicked.

Use the **System.Web.UI.WebControls.HyperLinkColumn.Target** property to specify the frame or window that displays the Web page linked to when a hyperlink in the column is clicked.

Text

TrackViewState

[C#] public virtual string Text {get; set;}

[C++] public: \_\_property virtual String\* get\_Text();public: \_\_property virtual void set Text(String\*);

1	[VB] Overridable Public Property Text As String
2	[JScript] public function get Text(): String; public function set Text(String);
3	
4	Description
5	Gets or sets the text caption to display for the hyperlinks in the column.
6	Use the System.Web.UI.WebControls.HyperLinkColumn.Text property
7	to specify the text caption to display for the hyperlinks in the column.
8	ViewState
9	Visible
10	FormatDataNavigateUrlValue
11	
12	[C#] protected virtual string FormatDataNavigateUrlValue(object dataUrlValue);
13	[C++] protected: virtual String* FormatDataNavigateUrlValue(Object*
14	dataUrlValue);
15	[VB] Overridable Protected Function FormatDataNavigateUrlValue(ByVal
16	dataUrlValue As Object) As String
17	[JScript] protected function FormatDataNavigateUrlValue(dataUrlValue : Object)
18	: String;
19	
20	Description
21	FormatDataTextValue
22	
23	[C#] protected virtual string FormatDataTextValue(object dataTextValue);
24	[C++] protected: virtual String* FormatDataTextValue(Object* dataTextValue);
25	[VB] Overridable Protected Function FormatDataTextValue(ByVal

1	dataTextValue As Object) As String
2	[JScript] protected function FormatDataTextValue(dataTextValue : Object) :
3	String;
4	
5	Description
6	Initialize
7	
8	[C#] public override void Initialize();
9	[C++] public: void Initialize();
10	[VB] Overrides Public Sub Initialize()
11	[JScript] public override function Initialize();
12	
13	Description
14	InitializeCell
15	
16	[C#] public override void InitializeCell(TableCell cell, int columnIndex,
17	ListItemType itemType);
18	[C++] public: void InitializeCell(TableCell* cell, int columnIndex, ListItemType
19	itemType);
20	[VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal
21	columnIndex As Integer, ByVal itemType As ListItemType)
22	[JScript] public override function InitializeCell(cell: TableCell, columnIndex: int
23	itemType : ListItemType);
24	
25	Description

Initializes the cell representing this column with the contained hyperlink. HyperLinks are created for all items in the DataGrid except the Header and 2 Footer items. The cell to be initialized. The index of the column that contains the 3 cell. The type of item that the cell is part of. HyperLinkControlBuilder class (System.Web.UI.WebControls) **TrackViewState** 8 Description 9 Interacts with the parser to build a 10 System.Web.UI.WebControls.HyperLink control. 11 To create a custom control builder for a 12 System.Web.UI.WebControls.HyperLink derived control, you need to inherit 13 from this class. 14 HyperLinkControlBuilder 15 Example Syntax: 16 **TrackViewState** 17 18 [C#] public HyperLinkControlBuilder(); 19 [C++] public: HyperLinkControlBuilder(); 20 [VB] Public Sub New() 21 [JScript] public function HyperLinkControlBuilder(); 22 ControlType 23 **FChildrenAsProperties** 24

FIsNonParserAccessor

1	HasAspCode
2	ID
3	InDesigner
4	NamingContainerType
5	Parser
6	TagName
7	AllowWhitespaceLiterals
8	
9	[C#] public override bool AllowWhitespaceLiterals();
10	[C++] public: bool AllowWhitespaceLiterals();
11	[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
12	[JScript] public override function AllowWhitespaceLiterals(): Boolean;
13	
14	Description
15	Gets a value that indicates whether white spaces are allowed in literals for
16	this control.
17	Return Value: Overloaded to always returns false to indicate that white spaces are
18	not allowed.
19	This method overrides the
20	System.Web.UI.ControlBuilder.AllowWhitespaceLiterals property to ignore
21	white space in the System.Web.UI.WebControls.HyperLink control.
22	Image class (System.Web.UI.WebControls)
23	ToString
24	
25	

Description

Displays an image on a Web page.

Use the **System.Web.UI.WebControls.Image** control to display an image on the Web page. The path to the displayed image is specified by setting the **System.Web.UI.WebControls.Image.ImageUrl** property. You can specify the text to display in place of image when the image is not available by setting the **System.Web.UI.WebControls.Image.AlternateText** property. The alignment of the image in relation to other elements on the Web page is specified by setting **System.Web.UI.WebControls.Image.ImageAlign** property.

Image

Example Syntax:

**ToString** 

[C#] public Image();

[C++] public: Image();

[VB] Public Sub New()

[JScript] public function Image();

Description

Initializes a new instance of the **System.Web.UI.WebControls.Image** class.

Use this constructor to create and initialize a new instance of the **System.Web.UI.WebControls.Image** class.

AccessKey AlternateText 2 **ToString** 3 5 Description 6 Gets or sets the alternate text displayed in the 7 System. Web.UI. WebControls. Image control when the image is unavailable. 8 Browsers that support the ToolTips feature display this text as a ToolTip. 9 Use this property to specify the text to display if the image specified in the 10 System.Web.UI.WebControls.Image.ImageUrl property is not available. In 11 browsers that support the ToolTips feature, this text also displays as a ToolTip. 12 Attributes 13 BackColor 14 BorderColor 15 BorderStyle 16 **BorderWidth** 17 ChildControlsCreated 18 ClientID 19 Context 20 Controls 21 ControlStyle 22 ControlStyleCreated 23 CssClass 24

Enabled

lee @hayes pilc 509+324+9256

25

1	ToString
2	
3	
4	Description
5	Gets or sets a value indicating whether the control is enabled.
6	This property is inherited from the
7	System.Web.UI.WebControls.WebControl class and is not applicable to the
8	System.Web.UI.WebControls.Image control.
9	EnableViewState
10	Events
11	Font
12	ToString
13	
14	
15	Description
16	Gets the font properties for the text associated with the control.
17	This property is inherited from the
18	System.Web.UI.WebControls.WebControl class and is not applicable to the
19	System.Web.UI.WebControls.Image control.
20	ForeColor
21	HasChildViewState
22	Height
23	ID
24	ImageAlign
25	ToString

Description
Gets

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets the alignment of the System.Web.UI.WebControls.Image

control in relation to other elements on the Web page.

Use the **System.Web.UI.WebControls.Image.ImageAlign** property to specify or determine the alignment of the image in relation to other elements on the Web page. The following table lists the possible alignments.

ImageUrl

**ToString** 

[C#] public virtual string ImageUrl {get; set;}

[C++] public: \_\_property virtual String\* get\_ImageUrl();public: \_\_property virtual void set ImageUrl(String\*);

[VB] Overridable Public Property ImageUrl As String

[JScript] public function get ImageUrl(): String; public function set ImageUrl(String);

Description

Gets or sets the location of an image to display in the System.Web.UI.WebControls.Image control.

Use the **System.Web.UI.WebControls.Image.ImageUrl** property to specify the URL of an image to display in the

**System.Web.UI.WebControls.Image** control. You can use a relative or an absolute URL. A relative URL relates the location of the image to the location of

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

the Web page without specifying a complete path on the server. The path is relative to the location of the Web page. This makes it easier to move the entire site to another directory on the server without updating the code. An absolute URL provides the complete path, so moving the site to another directory requires that you update the code. IsTrackingViewState NamingContainer Page Parent Site Style **TabIndex** TagKey TagName **TemplateSourceDirectory ToolTip** UniqueID ViewState ViewStateIgnoresCase Visible Width AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer); [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

2	HtmlTextWriter)
3	[JScript] protected override function AddAttributesToRender(writer:
4	HtmlTextWriter);
5	
6	Description
7	Adds the attributes of an System.Web.UI.WebControls.Image to the
8	output stream for rendering on the client. A System.Web.UI.HtmlTextWriter
9	that contains the output stream to render on the client browser.
10	RenderContents
11	
12	[C#] protected override void RenderContents(HtmlTextWriter writer);
13	[C++] protected: void RenderContents(HtmlTextWriter* writer);
14	[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)
15	[JScript] protected override function RenderContents(writer: HtmlTextWriter);
16	
17	Description
18	ImageAlign enumeration (System.Web.UI.WebControls)
19	TrackViewState
20	
21	
22	Description
23	Specifies the alignment of an image in relation to the text of a Web page.
24	The System.Web.UI.WebControls.ImageAlign enumeration represents
25	the alignment options for an image relative to the text of a Web page.

,	Tra	c	V	ev	27	ta	te
	111	UK	.ν.	ICV	V 1.7	La	L٧

1

3

4

5

6

7

8

9

10

12

13

14 15

16 17

18

19

20

22

23

24

25

[C#] public const ImageAlign AbsBottom;

[C++] public: const ImageAlign AbsBottom;

[VB] Public Const AbsBottom As ImageAlign

[JScript] public var AbsBottom : ImageAlign;

Description

The lower edge of the image is aligned with the lower edge of the largest element on the same line.

TrackViewState

[C#] public const ImageAlign AbsMiddle;

[C++] public: const ImageAlign AbsMiddle;

[VB] Public Const AbsMiddle As ImageAlign

[JScript] public var AbsMiddle: ImageAlign;

Description

The middle of the image is aligned with the middle of the largest element on the same line.

**TrackViewState** 

[C#] public const ImageAlign Baseline;

[C++] public: const ImageAlign Baseline;

[VB] Public Const Baseline As ImageAlign

1	[JScript] public var Baseline : ImageAlign;
2	
3	Description
4	The lower edge of the image is aligned with the lower edge of the first line
5	of text.
6	TrackViewState
7	
8	[C#] public const ImageAlign Bottom;
9	[C++] public: const ImageAlign Bottom;
10	[VB] Public Const Bottom As ImageAlign
11	[JScript] public var Bottom : ImageAlign;
12	
13	Description
14	The lower edge of the image is aligned with the lower edge of the first line
15	of text.
16	TrackViewState
17	
18	[C#] public const ImageAlign Left;
19	[C++] public: const ImageAlign Left;
20	[VB] Public Const Left As ImageAlign
21	[JScript] public var Left : ImageAlign;
22	
23	Description
24	The image is aligned on the left edge of the Web page with text wrapping
25	on the right.

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Trac	νVi	PIXIS	tate
Traci	KVI	CWS	iaic

[C#] public const ImageAlign Middle;

[C++] public: const ImageAlign Middle;

[VB] Public Const Middle As ImageAlign

[JScript] public var Middle: ImageAlign;

Description

The middle of the image is aligned with the lower edge of the first line of text.

TrackViewState

[C#] public const ImageAlign NotSet;

[C++] public: const ImageAlign NotSet;

[VB] Public Const NotSet As ImageAlign

[JScript] public var NotSet : ImageAlign;

Description

The alignment is not set.

**TrackViewState** 

[C#] public const ImageAlign Right;

[C++] public: const ImageAlign Right;

[VB] Public Const Right As ImageAlign

[JScript] public var Right : ImageAlign;

Description

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

The image is aligned on the right edge of the Web page with text wrapping on the left.

TrackViewState

[C#] public const ImageAlign TextTop;

[C++] public: const ImageAlign TextTop;

[VB] Public Const TextTop As ImageAlign

[JScript] public var TextTop : ImageAlign;

Description

The upper edge of the image is aligned with the upper edge of the highest text on the same line.

TrackViewState

[C#] public const ImageAlign Top;

[C++] public: const ImageAlign Top;

[VB] Public Const Top As ImageAlign

[JScript] public var Top: ImageAlign;

Description

The upper edge of the image is aligned with the upper edge of the highest element on the same line.

ImageButton class (System.Web.UI.WebControls)

1	ToString
2	
3	
4	Description
5	A control that displays an image and responds to mouse clicks on the
6	image.
7	Use the System.Web.UI.WebControls.ImageButton control to display an
8	image that responds to mouse clicks.
9	ImageButton
10	Example Syntax:
11	ToString
12	
13	[C#] public ImageButton();
14	[C++] public: ImageButton();
15	[VB] Public Sub New()
16	[JScript] public function ImageButton();
17	
18	Description
19	Initializes a new instance of the
20	System.Web.UI.WebControls.ImageButton class.
21	Use this constructor to create and initialize a new instance of the
22	System.Web.UI.WebControls.ImageButton class.
23	AccessKey
24	AlternateText
25	Attributes

17

18

19

20

21

22

23

24

Gets or sets a value indicating whether validation is performed when the

 ${\bf System. Web. UI. WebControls. Image Button}\ control\ is\ clicked.$ 

By default, page validation is performed when a

System.Web.UI.WebControls.ImageButton control is clicked. Page validation determines whether the input controls associated with a validation control on the page all pass the validation rules specified by the validation control.

ChildControlsCreated

ClientID

CommandArgument

**ToString** 

Description

Gets or sets an optional argument that provides additional information about the **System.Web.UI.WebControls.ImageButton.CommandName** property.

1	Sometimes, multiple System.Web.UI.WebControls.ImageButton
2	controls are related and share the same value for the
3	System.Web.UI.WebControls.ImageButton.CommandName property, such as
4	Sort . Use this property to supplement the
5	System.Web.UI.WebControls.ImageButton.CommandName property with
6	additional information about the command to perform, such as Ascending. The
7	values of the System. Web. UI. Web Controls. Image Button. Command Name and
8	System.Web.UI.WebControls.ImageButton.CommandArgument properties
9	are typically used in the
10	System.Web.UI.WebControls.ImageButton.OnCommand(System.Web.UI.W
11	ebControls.CommandEventArgs) event handler to determine the action to
12	perform when the System.Web.UI.WebControls.ImageButton control is
13	clicked.
14	CommandName
15	ToString
16	
17	[C#] public string CommandName {get; set;}
18	[C++] public:property String* get_CommandName();public:property void
19	set_CommandName(String*);
20	[VB] Public Property CommandName As String
21	[JScript] public function get CommandName(): String;public function set
22	CommandName(String);
23	
24	Description
25	

24

25

Height

ID

Gets or sets the command name associated with the  ${\bf System. Web. UI. Web Controls. Image Button\ control.}$ Use this property to specify the command to perform when the 3  ${\bf System. Web. UI. WebControls. Image Button}\ \ {\bf control}\ \ {\bf is}\ \ {\bf clicked},\ {\bf such}\ \ {\bf as}\ \ {\bf Sort}\ ,$ Cancel, and Edit. This allows multiple 5 System.Web.UI.WebControls.ImageButton controls to be placed on the same Web page. The value in this property can then be programmatically identified in 7 the 8 System. Web. UI. WebControls. Image Button. On Command (System. Web. UI. WebControls.)9 ebControls.CommandEventArgs) event handler to determine the appropriate 10 action to perform when each System. Web. UI. Web Controls. Image Button 11 control is clicked. 12 Context 13 Controls 14 ControlStyle 15 ControlStyleCreated 16 CssClass 17 Enabled 18 **EnableViewState** 19 **Events** 20 Font 21 ForeColor 22 HasChildViewState

ImageAlign
ImageUrl
IsTrackingViewState
NamingContainer
Page
Parent
Site
Style
TabIndex
TagKey
ToString

# Description

Gets a value that represents the tag HtmlTextWriterTag.Input. This property is read-only.

Overrides the base implementation and returns the tag

HtmlTextWriterTag.Input instead of the default HtmlTextWriterTag.Img Gets a

value that represents the tag HtmlTextWriterTag.Input. This property is read-only.

TagName

TemplateSourceDirectory

ToolTip

UniqueID

ViewState

ViewStateIgnoresCase

Visible Width 2 **ToString** 3 5 Description 6 Occurs when the System. Web. UI. Web Controls. Image Button is clicked. 7 The System.Web.UI.WebControls.ImageButton.Click event is raised 8 when the System. Web. UI. WebControls. ImageButton control is clicked. 9 **ToString** 10 11 [C#] public event CommandEventHandler Command; 12 [C++] public: \_\_event CommandEventHandler\* Command; 13 [VB] Public Event Command As CommandEventHandler 14 15 Description 16 Occurs when the System. Web. UI. Web Controls. Image Button is clicked. 17 The System.Web.UI.WebControls.ImageButton.Command event is 18 raised when an System. Web.UI. WebControls. ImageButton control is clicked. 19 AddAttributesToRender 20 21 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer); 22 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer); 23 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As 24 HtmlTextWriter) 25

[JScript] protected override function AddAttributesToRender(writer: HtmlTextWriter);

Description

Adds the attributes of an **System.Web.UI.WebControls.ImageButton** to the output stream for rendering on the client. The output stream to render on the client.

OnClick

[C#] protected virtual void OnClick(ImageClickEventArgs e);
 [C++] protected: virtual void OnClick(ImageClickEventArgs\* e);
 [VB] Overridable Protected Sub OnClick(ByVal e As ImageClickEventArgs)
 [JScript] protected function OnClick(e : ImageClickEventArgs);

## Description

Raises the **System.Web.UI.WebControls.ImageButton.Click** event and allows you to handle the **System.Web.UI.WebControls.ImageButton.Click** event directly.

The System.Web.UI.WebControls.ImageButton.Click event is raised when the System.Web.UI.WebControls.ImageButton control is clicked. By using the

System.Web.UI.WebControls.ImageButton.OnClick(System.Web.UI.ImageC lickEventArgs) event handler, you can programmatically determine the coordinates where the image is clicked. You can then code a response, based on the values of these coordinates. Note the origin (0, 0) is located at the upper left

corner of the image. A **System.Web.UI.ImageClickEventArgs** that contains the event data.

**OnCommand** 

[C#] protected virtual void OnCommand(CommandEventArgs e);

[C++] protected: virtual void OnCommand(CommandEventArgs\* e);

[VB] Overridable Protected Sub OnCommand(ByVal e As CommandEventArgs)

[JScript] protected function OnCommand(e: CommandEventArgs);

### Description

Raises the **System.Web.UI.WebControls.ImageButton.Command** event and allows you to handle the

System.Web.UI.WebControls.ImageButton.Command event directly.

The System.Web.UI.WebControls.ImageButton.Command event is raised when the System.Web.UI.WebControls.ImageButton control is clicked. The

System.Web.UI.WebControls.ImageButton.OnCommand(System.Web.UI.WebControls.CommandEventArgs) event handler is used to make the System.Web.UI.WebControls.ImageButton control behave like a command button. A command name can be associated with the control by using the System.Web.UI.WebControls.ImageButton.CommandName property. This allows multiple System.Web.UI.WebControls.ImageButton controls to be placed on the Web page. The value in this property can then be programmatically identified in the

System. Web. UI. Web Controls. Image Button. On Command (System. Web. UI.

ebControls.CommandEventArgs) event handler to determine the appropriate
action to perform when each System.Web.UI.WebControls.ImageButton
control is clicked. The
${\bf System. Web. UI. Web Controls. Image Button. Command Argument \ property \ can}$
also be used to pass additional information about the command, such as specifying
ascending order. A System.Web.UI.WebControls.CommandEventArgs that
contains the event data.
OnPreRender
[C#] protected override void OnPreRender(EventArgs e);
[C++] protected: void OnPreRender(EventArgs* e);
[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)
[JScript] protected override function OnPreRender(e : EventArgs);
Description
Determine if the image has been clicked prior to rendering on the client.
IPostBackDataHandler.LoadPostData
[C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
NameValueCollection postCollection);
[C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
NameValueCollection* postCollection);
[VB] Function LoadPostData(ByVal postDataKey As String, ByVal
postCollection As NameValueCollection) As Boolean Implements
IPostBackDataHandler.LoadPostData

1	[JScript] function [PostBackDataHandler.LoadPostData(postDataRey . String,
2	postCollection : NameValueCollection) : Boolean;
3	IPostBackDataHandler.RaisePostDataChangedEvent
4	
5	[C#] void IPostBackDataHandler.RaisePostDataChangedEvent();
6	[C++] void IPostBackDataHandler::RaisePostDataChangedEvent();
7	[VB] Sub RaisePostDataChangedEvent() Implements
8	IPostBackDataHandler.RaisePostDataChangedEvent
9	[JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();
10	IPostBackEventHandler.RaisePostBackEvent
11	•
12	[C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);
13	[C++] void IPostBackEventHandler::RaisePostBackEvent(String*
14	eventArgument);
15	[VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements
16	IPostBackEventHandler.RaisePostBackEvent
17	[JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument
18	String);
19	IRepeatInfoUser interface (System.Web.UI.WebControls)
20	TrackViewState
21	
22	
23	Description
24	Specifies a contract for implementing
25	System.Web.UI.WebControls.Repeater objects in list controls.

HasFooter **TrackViewState** 3 [C#] bool HasFooter {get;} [C++] bool get\_HasFooter(); [VB] ReadOnly Property HasFooter As Boolean [JScript] abstract function get HasFooter(): Boolean; 7 8 Description 9 Indicates whether the Repeater contains a footer item. 10 HasHeader 11 **TrackViewState** 12 13 [C#] bool HasHeader {get;} 14 [C++] bool get\_HasHeader(); 15 [VB] ReadOnly Property HasHeader As Boolean 16 [JScript] abstract function get HasHeader() : Boolean; 17 18 Description 19 Indicates whether the System.Web.UI.WebControls.Repeater contains a 20 header item. 21 HasSeparators 22 TrackViewState 23 24 [C#] bool HasSeparators {get;}

1	[C++] bool get_HasSeparators();
2	[VB] ReadOnly Property HasSeparators As Boolean
3	[JScript] abstract function get HasSeparators(): Boolean;
4	
5	Description
6	Indicates whether the Repeater contains separator items.
7	RepeatedItemCount
8	TrackViewState
9	
10	[C#] int RepeatedItemCount {get;}
11	[C++] int get_RepeatedItemCount();
12	[VB] ReadOnly Property RepeatedItemCount As Integer
13	[JScript] abstract function get RepeatedItemCount(): int;
14	
15	Description
16	Specifies the item count of the Repeater.
17	GetItemStyle
18	
19	[C#] Style GetItemStyle(ListItemType itemType, int repeatIndex);
20	[C++] Style* GetItemStyle(ListItemType itemType, int repeatIndex);
21	[VB] Function GetItemStyle(ByVal itemType As ListItemType, ByVal
22	repeatIndex As Integer) As Style
23	[JScript] function GetItemStyle(itemType : ListItemType, repeatIndex : int) :
24	Style;
25	

### Description

Retrieves the item style with the specified item type and location within the System.Web.UI.WebControls.Repeater.

Return Value: A System.Web.UI.WebControls.Style that represents the Repeater item style. A System.Web.UI.WebControls.ListItemType that represents the specified type of the System.Web.UI.WebControls.Repeater item. An ordinal index that specifies the location of the item within the

# System.Web.UI.WebControls.Repeater.

### RenderItem

[C#] void RenderItem(ListItemType itemType, int repeatIndex, RepeatInfo repeatInfo, HtmlTextWriter writer);
[C++] void RenderItem(ListItemType itemType, int repeatIndex, RepeatInfo\*

repeatInfo, HtmlTextWriter\* writer);

[VB] Sub RenderItem(ByVal itemType As ListItemType, ByVal repeatIndex As Integer, ByVal repeatInfo As RepeatInfo, ByVal writer As HtmlTextWriter)

[JScript] function RenderItem(itemType : ListItemType, repeatIndex : int, repeatInfo : RepeatInfo, writer : HtmlTextWriter);

Description

Renders the System.Web.UI.WebControls.Repeater item with the specified information. A System.Web.UI.WebControls.ListItemType that represents the specified type of the System.Web.UI.WebControls.Repeater item. An ordinal index that specifies the location of the item within the

System.Web.UI.WebControls.Repeater. A System.Web.UI.WebControls.RepeatInfo that represents the information used 2 to render items using a System. Web.UI. WebControls. Repeater . The output 3 stream that renders HTML content to the client. Label class (System.Web.UI.WebControls) RenderItem 8 Description 9 Represents a label control, which displays text on a Web page. 10 Use the System. Web. UI. Web Controls. Label control to display text in a 11 set location on the page. Unlike static text, you can customize the displayed text 12 through the System.Web.UI.WebControls.Label.Text property. 13 Label 14 Example Syntax: 15 RenderItem 16 17 [C#] public Label(); 18 [C++] public: Label(); 19 [VB] Public Sub New() 20 [JScript] public function Label(); Initializes a new instance of the 21 System.Web.UI.WebControls.Label class. 22 23 Description 24 25

Initializes a new instance of the System. Web. UI. Web Controls. Label class. 2 Use this constructor to create and initialize a new instance of the 3  $System. Web. UI. Web Controls. Label \ class.$ AccessKey 5 Attributes BackColor BorderColor BorderStyle 9 BorderWidth 10 ChildControlsCreated 11 ClientID 12 Context 13 Controls 14 ControlStyle 15 ControlStyleCreated 16 CssClass 17 Enabled 18 EnableViewState 19 **Events** 20 Font 21 ForeColor 22 HasChildViewState 23 Height 24 ID 25

IsTrackingViewState
NamingContainer
Page
Parent
Site
Style
TabIndex
TagKey
TagName
TemplateSourceDirectory
Text
RenderItem

Description

Gets or sets the text content of the **System.Web.UI.WebControls.Label** control.

Use the **System.Web.UI.WebControls.Label.Text** property to specify or determine the text content of the **System.Web.UI.WebControls.Label** control. This property is commonly used to programmatically customize the text that is displayed in the **System.Web.UI.WebControls.Label** control.

ToolTip

UniqueID

ViewState

ViewStateIgnoresCase

Visible Width 2 AddParsedSubObject 3 [C#] protected override void AddParsedSubObject(object obj); 5 [C++] protected: void AddParsedSubObject(Object\* obj); 6 [VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object) 7 [JScript] protected override function AddParsedSubObject(obj : Object); 8 9 Description 10 LoadViewState 11 12 [C#] protected override void LoadViewState(object savedState); 13 [C++] protected: void LoadViewState(Object\* savedState); 14 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object) 15 [JScript] protected override function LoadViewState(savedState : Object); 16 17 Description 18 Load previously saved state. Overridden to synchronize Text property with 19 LiteralContent. Represents the previously saved state. 20 RenderContents 21 22 [C#] protected override void RenderContents(HtmlTextWriter writer); 23 [C++] protected: void RenderContents(HtmlTextWriter\* writer); 24 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

[JScript] protected override function RenderContents(writer: HtmlTextWriter); 2 Description 3 Renders the contents of the System.Web.UI.WebControls.Label into the specified writer. The output stream that renders HTML content to the client. 5 LabelControlBuilder class (System.Web.UI.WebControls) 6 **TrackViewState** 9 Description 10 Interacts with the parser to build a System.Web.UI.WebControls.Label 11 control. 12 To create a custom control builder for a 13 System.Web.UI.WebControls.Label derived control, you need to inherit from 14 this class. 15 LabelControlBuilder 16 Example Syntax: 17 **TrackViewState** 18 19 [C#] public LabelControlBuilder(); 20 [C++] public: LabelControlBuilder(); 21 [VB] Public Sub New() 22 [JScript] public function LabelControlBuilder(); 23 ControlType 24 **FChildrenAsProperties** 25

FIsNonParserAccessor HasAspCode 2 ID InDesigner NamingContainerType Parser **TagName** AllowWhitespaceLiterals 8 9 [C#] public override bool AllowWhitespaceLiterals(); 10 [C++] public: bool AllowWhitespaceLiterals(); 11 [VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean 12 [JScript] public override function AllowWhitespaceLiterals(): Boolean; 13 14 Description 15 Specifies whether white space literals are allowed. 16 Return Value: false for all cases. 17 This method overrides the 18 System.Web.UI.ControlBuilder.AllowWhitespaceLiterals property to ignore 19 white space in the **System.Web.UI.WebControls.Label** control. 20 LinkButton class (System.Web.UI.WebControls) 21 **ToString** 22 23 24 Description

3

5

9

10

11

12

13

14

15

16

17

18

19

20

25

Displays a hyperlink style button control on a Web page. Use the System. Web. UI. WebControls. Link Button control to create a hyperlink style button on the Web page. You can create either a submit button or a command button. LinkButton Example Syntax: **ToString** [C#] public LinkButton(); [C++] public: LinkButton(); [VB] Public Sub New() [JScript] public function LinkButton(); Description Initializes a new instance of the System.Web.UI.WebControls.LinkButton class. Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.LinkButton control. AccessKey Attributes BackColor

BackColor
BorderColor
BorderStyle
BorderWidth

CausesValidation

**ToString** 

Description

Gets or sets a value indicating whether validation is performed when the **System.Web.UI.WebControls.LinkButton** control is clicked.

By default, page validation is performed when a 
System.Web.UI.WebControls.LinkButton control is clicked. Page validation determines whether the input controls associated with a validation control on the page all pass the validation rules specified by the validation control.

ChildControlsCreated

ClientID

CommandArgument

**ToString** 

Description

Gets or sets an optional argument passed to the

System.Web.UI.WebControls.LinkButton.Command event handler along with the associated System.Web.UI.WebControls.LinkButton.CommandName property.

Use the **System.Web.UI.WebControls.LinkButton.CommandArgument** property to specify an argument that complements the

 ${\bf System. Web. UI. WebControls. Link Button. Command Name\ property.}$ 

CommandName

_	. •		
 $\sim$	+	*	^
'oS	111		v
 $\sim$	CLI		,-

[C#] public string CommandName {get; set;}

[C++] public: \_\_property String\* get\_CommandName();public: \_\_property void set CommandName(String\*);

[VB] Public Property CommandName As String

[JScript] public function get CommandName() : String;public function set CommandName(String);

#### Description

Gets or sets the command name associated with the System.Web.UI.WebControls.LinkButton control. This value is passed to the System.Web.UI.WebControls.LinkButton.Command event handler along with the System.Web.UI.WebControls.LinkButton.CommandArgument property.

When you have multiple **System.Web.UI.WebControls.LinkButton** controls on a Web page, use the

System.Web.UI.WebControls.LinkButton.CommandName property to specify or determine the command name associated with the

System.Web.UI.WebControls.LinkButton control clicked. You can set the System.Web.UI.WebControls.LinkButton.CommandName property with any string that identifies the command to perform. You can then programmatically determine the command name of the System.Web.UI.WebControls.LinkButton control and perform the appropriate actions.

Context

Controls

lee@hayes plic 509-324-9256 1942

MS1-863US.APP

1	ControlStyle
2	ControlStyleCreated
3	CssClass
4	Enabled
5	EnableViewState
6	Events
7	Font
8	ForeColor
9	HasChildViewState
10	Height
11	ID
12	IsTrackingViewState
13	NamingContainer
14	Page
15	Parent
16	Site
17	Style
18	TabIndex
19	TagKey
20	TagName
21	TemplateSourceDirectory
22	Text
23	ToString
24	

Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the text caption displayed on the

System.Web.UI.WebControls.LinkButton control.

Use the **System.Web.UI.WebControls.LinkButton.Text** property to specify or determine the caption to display on the

System.Web.UI.WebControls.LinkButton control.

**ToolTip** 

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

**ToString** 

Description

Occurs when the **System.Web.UI.WebControls.LinkButton** control is clicked.

The System.Web.UI.WebControls.LinkButton.Click event is raised when the System.Web.UI.WebControls.LinkButton control is clicked. This event is commonly used when no command name is associated with the System.Web.UI.WebControls.LinkButton control, such as a submit button.

**ToString** 

[C#] public event CommandEventHandler Command;[C++] public: \_\_event CommandEventHandler\* Command;[VB] Public Event Command As CommandEventHandler

### Description

Occurs when the System. Web. UI. Web Controls. Button control is clicked.

The System.Web.UI.WebControls.LinkButton.Command event is raised when the System.Web.UI.WebControls.LinkButton control is clicked. This event is commonly used when a command name, such as Sort, is associated with the System.Web.UI.WebControls.LinkButton control. This allows you to create multiple System.Web.UI.WebControls.LinkButton controls on a Web page and programmatically determine which System.Web.UI.WebControls.LinkButton control is clicked.

#### AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);
[C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);
[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As
HtmlTextWriter)

[JScript] protected override function AddAttributesToRender(writer:
HtmlTextWriter);

# Description

Render the attributes on the begin tag.

AddParsedSubObje
AddParsedSubObje

C#	protected	override	void	AddPars	edSubOl	oiect	(ob	iect	obi	i):
$1 \cup \pi$	DIOLOCIOA	Override	Y OLU	z xuur aro	Casasco	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	( O O	CCC	00	,

[C++] protected: void AddParsedSubObject(Object\* obj);

[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)

[JScript] protected override function AddParsedSubObject(obj : Object);

Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

LoadViewState

[C#] protected override void LoadViewState(object savedState);

[C++] protected: void LoadViewState(Object\* savedState);

[VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)

[JScript] protected override function LoadViewState(savedState : Object);

# Description

Load previously saved state. Overridden to synchronize Text property with LiteralContent.

**OnClick** 

[C#] protected virtual void OnClick(EventArgs e);

[C++] protected: virtual void OnClick(EventArgs\* e);

[VB] Overridable Protected Sub OnClick(ByVal e As EventArgs)

[JScript] protected function OnClick(e: EventArgs);

Description

Raises the **System.Web.UI.WebControls.LinkButton.Click** event of the **System.Web.UI.WebControls.LinkButton** control.

The System.Web.UI.WebControls.LinkButton.Click event is raised when the System.Web.UI.WebControls.LinkButton control is clicked. This event is commonly used when no command name is associated with the System.Web.UI.WebControls.LinkButton control, such as a submit button. A System.EventArgs that contains the event data.

OnCommand

[C#] protected virtual void OnCommand(CommandEventArgs e);
[C++] protected: virtual void OnCommand(CommandEventArgs\* e);
[VB] Overridable Protected Sub OnCommand(ByVal e As CommandEventArgs)
[JScript] protected function OnCommand(e : CommandEventArgs);

#### Description

Raises the **System.Web.UI.WebControls.LinkButton.Command** event of the **System.Web.UI.WebControls.LinkButton** control.

The System.Web.UI.WebControls.LinkButton.Command event is raised when the System.Web.UI.WebControls.LinkButton control is clicked. This event is commonly used when a command name, such as Sort, is associated with the System.Web.UI.WebControls.LinkButton control. This allows you to create multiple System.Web.UI.WebControls.LinkButton controls on a Web page and programmatically determine which System.Web.UI.WebControls.LinkButton

1	control is clicked. A System.Web.UI.WebControls.CommandEventArgs that
2	contains the event data.
3	RenderContents
4	
5	[C#] protected override void RenderContents(HtmlTextWriter writer);
6	[C++] protected: void RenderContents(HtmlTextWriter* writer);
7	[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)
8	[JScript] protected override function RenderContents(writer: HtmlTextWriter);
9	
10	Description
11	The output stream that renders HTML content to the client.
12	IPostBackEventHandler.RaisePostBackEvent
13	
14	[C#] void IPostBackEventHandler.RaisePostBackEvent(string eventArgument);
15	[C++] void IPostBackEventHandler::RaisePostBackEvent(String*
16	eventArgument);
17	[VB] Sub RaisePostBackEvent(ByVal eventArgument As String) Implements
18	IPostBackEventHandler.RaisePostBackEvent
19	[JScript] function IPostBackEventHandler.RaisePostBackEvent(eventArgument :
20	String);
21	LinkButtonControlBuilder class (System.Web.UI.WebControls)
22	TrackViewState
23	
24	
25	Description

1	Interacts with the parser to build a
2	System.Web.UI.WebControls.LinkButton control.
3	To create a custom control builder for a
4	System.Web.UI.WebControls.LinkButton derived control, you need to inherit
5	from this class.
6	LinkButtonControlBuilder
7	Example Syntax:
8	TrackViewState
9	
10	[C#] public LinkButtonControlBuilder();
11	[C++] public: LinkButtonControlBuilder();
12	[VB] Public Sub New()
13	[JScript] public function LinkButtonControlBuilder();
14	ControlType
15	FChildrenAsProperties
16	FIsNonParserAccessor
17	HasAspCode
18	ID
19	InDesigner
20	NamingContainerType
21	Parser
22	TagName
23	AllowWhitespaceLiterals
24	
25	[C#] public override bool AllowWhitespaceLiterals();

1	[C++] public: bool AllowWhitespaceLiterals();
2	[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
3	[JScript] public override function AllowWhitespaceLiterals(): Boolean;
4	
5	Description
6	Specifies whether white space literals are allowed.
7	Return Value: false for all cases.
8	This method overrides
9	System.Web.UI.ControlBuilder.AllowWhitespaceLiterals to ignore white
10	space in the System.Web.UI.WebControls.LinkButton control.
11	ListBox class (System.Web.UI.WebControls)
12	ToString
13	
14	
15	Description
16	Represents a list box control that allows single or multiple item selection.
17	Use the System.Web.UI.WebControls.ListBox control to create a list
18	control that allows single or multiple item selection. Use the
19	System.Web.UI.WebControls.ListBox.Rows property to specify the height of
20	the control. To enable multiple item selection, set the
21	System.Web.UI.WebControls.ListBox.SelectionMode property to
22	ListSelectionMode.Multiple .
23	ListBox
24	Example Syntax:
25	ToString

[C#] public ListBox(); [C++] public: ListBox(); [VB] Public Sub New() [JScript] public function ListBox(); 5 6 Description Initializes a new instance of the System. Web.UI. WebControls. ListBox 8 class. 9 Use this constructor to create and initialize a new instance of the 10 System.Web.UI.WebControls.ListBox class. 11 AccessKey 12 Attributes 13 AutoPostBack 14 BackColor 15 BorderColor 16 **ToString** 17 18 19 Description 20 Gets or sets the border color of the control. 21 The  ${\bf System.Web.UI.WebControls.ListBox.BorderColor}$  property is 22  $inherited \ from \ the \ \textbf{System.Web.UI.WebControls.WebControl} \ class \ and \ is \ not$ 23 applicable to the System.Web.UI.WebControls.ListBox control. 24

BorderStyle

25

1	ToString
2	
3	[C#] public override BorderStyle BorderStyle {get; set;}
4	[C++] public:property virtual BorderStyle get_BorderStyle();public:
5	property virtual void set_BorderStyle(BorderStyle);
6	[VB] Overrides Public Property BorderStyle As BorderStyle
7	[JScript] public function get BorderStyle() : BorderStyle; public function set
8	BorderStyle(BorderStyle);
9	
10	Description
11	Gets or sets the border style of the control.
12	The System.Web.UI.WebControls.ListBox.BorderStyle property is
13	inherited from the System.Web.UI.WebControls.WebControl class and is not
14	applicable to the System.Web.UI.WebControls.ListBox control.
15	BorderWidth
16	ToString
17	
18	[C#] public override Unit BorderWidth {get; set;}
19	[C++] public:property virtual Unit get_BorderWidth();public:property
20	virtual void set_BorderWidth(Unit);
21	[VB] Overrides Public Property BorderWidth As Unit
22	[JScript] public function get BorderWidth(): Unit;public function set
23	BorderWidth(Unit);
24	
25	Description

2 3 5 ClientID Context Controls ControlStyle ControlStyleCreated 10 CssClass 11 DataMember 12 **DataSource** 13 DataTextField 14 15 DataValueField 16 Enabled 17 EnableViewState 18 **Events** 19 Font 20 ForeColor 21 HasChildViewState 22 Height 23

ID

IsTrackingViewState

24

Gets or sets the border width for the control.  $The \ {\bf System. Web. UI. WebControls. List Box. Border Width \ property \ is \\$ inherited from the System. Web.UI. WebControls. WebControl class and is not  $applicable \ to \ the \ {\bf System. Web. UI. WebControls. List Box} \ control.$ ChildControlsCreated DataTextFormatString

Items

NamingContainer

Page

Parent

Rows

**ToString** 

Description

Gets or sets the number of rows displayed in the

System.Web.UI.WebControls.ListBox control.

Use the **System.Web.UI.WebControls.ListBox.Rows** property to specify the number of rows to display in the **System.Web.UI.WebControls.ListBox** control.

SelectedIndex

SelectedItem

SelectionMode

**ToString** 

Description

23

24

25

Gets or sets the selection mode of the

System.Web.UI.WebControls.ListBox control.

Use the **System.Web.UI.WebControls.ListBox.SelectionMode** property to specify the mode behavior of the **System.Web.UI.WebControls.ListBox** 

3

4

5

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Width

AddAttributesToRender

control. Setting this property to ListSelectionMode.Single indicates only a single item can be selected from the System. Web.UI. WebControls. ListBox control, while ListSelectionMode.Multiple specifies multiple items can be selected. Site Style **TabIndex TagKey** TagName **TemplateSourceDirectory ToolTip ToString** Description Gets or sets the ToolTip text displayed when the mouse pointer rests over the control. The System.Web.UI.WebControls.ListBox.ToolTip property is inherited  $from \ the \ \textbf{System.Web.UI.WebControls.WebControl} \ class \ and \ is \ not \ applicable$ to the  $System.Web.UI.WebControls.ListBox\ control.$ UniqueID ViewState ViewStateIgnoresCase Visible

11	
1	
2	[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);
3	[C++] protected: void AddAttributesToRender(HtmlTextWriter* writer);
4	[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As
5	HtmlTextWriter)
6	[JScript] protected override function AddAttributesToRender(writer:
7	HtmlTextWriter);
8	
9	Description
10	Adds name, size, multiple, and onchange to list of attributes to render. The
11	output stream that renders HTML content to the client.
12	OnPreRender
13	
14	[C#] protected override void OnPreRender(EventArgs e);
15	[C++] protected: void OnPreRender(EventArgs* e);
16	[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)
17	[JScript] protected override function OnPreRender(e : EventArgs);
18	
19	Description
20	RenderContents
21	
22	[C#] protected override void RenderContents(HtmlTextWriter writer);
23	[C++] protected: void RenderContents(HtmlTextWriter* writer);
24	[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)
25	[JScript] protected override function RenderContents(writer: HtmlTextWriter);

1 Description IPostBackDataHandler.LoadPostData 3 [C#] bool IPostBackDataHandler.LoadPostData(string postDataKey, 5 NameValueCollection postCollection); [C++] bool IPostBackDataHandler::LoadPostData(String\* postDataKey, NameValueCollection\* postCollection); [VB] Function LoadPostData(ByVal postDataKey As String, ByVal 9 postCollection As NameValueCollection) As Boolean Implements 10 IPostBackDataHandler.LoadPostData 11 [JScript] function IPostBackDataHandler.LoadPostData(postDataKey: String, 12 postCollection: NameValueCollection): Boolean; 13 IPostBackDataHandler. RaisePostDataChangedEvent14 15 [C#] void IPostBackDataHandler.RaisePostDataChangedEvent(); 16 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent(); 17 [VB] Sub RaisePostDataChangedEvent() Implements 18 IPostBackDataHandler. RaisePostDataChangedEvent19  $[JScript]\ function\ IPostBackDataHandler. RaisePostDataChangedEvent();$ 20 ListControl class (System.Web.UI.WebControls) 21 TrackViewState 22 23 24

Description

Serves as the abstract base class that defines the properties, methods, and
events common for all list-type controls.
The System.Web.UI.WebControls.ListControl class is typically not
instantiated. Instead, to provide common basic functionality, this class is inherited
by others, such as the System.Web.UI.WebControls.CheckBoxList,
System.Web.UI.WebControls.DropDownList,
System.Web.UI.WebControls.ListBox, and
System. Web. UI. Web Controls. Radio Button List.
ListControl
Example Syntax:
TrackViewState
[C#] public ListControl();
[C++] public: ListControl();
[VB] Public Sub New()
[JScript] public function ListControl();
Description
Initializes a new instance of the
System.Web.UI.WebControls.ListControl class.

AccessKey

Attributes

AutoPostBack

TrackViewState

13

14

15

16

17

18

19

20

21

22

24

25

1

2

3

Description

Gets or sets a value indicating whether a postback to the server automatically occurs when the user changes the list selection.

Set this property to **true** if the server needs to capture the selection as soon as it is made. For example, other controls on the Web page can be automatically filled depending on the user's selection from a list control.

BackColor

BorderColor

BorderStyle

BorderWidth

ChildControlsCreated

ClientID

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

DataMember

TrackViewState

23

Description

Gets or sets the specific table in the 1 System.Web.UI.WebControls.ListControl.DataSource to bind to the control. 2 If the System. Web. UI. Web Controls. List Control. Data Source contains 3 more than one table, use this property to specify the exact table to bind to the control. 5 DataSource 6 **TrackViewState** 7 8 [C#] public virtual object DataSource {get; set;} 9 [C++] public: \_\_property virtual Object\* get\_DataSource();public: \_\_property 10 virtual void set\_DataSource(Object\*); 11 [VB] Overridable Public Property DataSource As Object 12 [JScript] public function get DataSource(): Object;public function set 13 DataSource(Object); 14 15 Description 16 Gets or sets the data source that populates the items of the list control. 17 Use this property to specify a source of data to populate a list control. 18 DataTextField 19 **TrackViewState** 20 21 [C#] public virtual string DataTextField {get; set;} 22 [C++] public: \_\_property virtual String\* get\_DataTextField();public: \_\_property 23 virtual void set DataTextField(String\*); 24 [VB] Overridable Public Property DataTextField As String

	[JScript] public function get DataTextField(): String;public function set	
	DataTextField(String);	
	Description	
	Gets or sets the field of the data source that provides the text content of the	
	list items.	
	Use this property to specify a field in the	
	System.Web.UI.WebControls.ListControl.DataSource to display as the items	
	of the list in a list control.	
	DataTextFormatString	
	TrackViewState	
	[C#] public virtual string DataTextFormatString {get; set;}	
	[C++] public:property virtual String* get_DataTextFormatString();public:	
	property virtual void set_DataTextFormatString(String*);	
ĺ	[VB] Overridable Public Property DataTextFormatString As String	
l	[JScript] public function get DataTextFormatString(): String;public function set	
	DataTextFormatString(String);	
	Description	
	Gets or sets the formatting string used to control how data bound to the list	
	control is displayed.	
	Use this property to provide a custom display format for the items in the list	
	control.	
	DataValueField	

24

25

## TrackViewState 1 2 [C#] public virtual string DataValueField {get; set;} 3 [C++] public: \_\_property virtual String\* get\_DataValueField();public: \_\_property virtual void set\_DataValueField(String\*); 5 [VB] Overridable Public Property DataValueField As String [JScript] public function get DataValueField(): String; public function set 7 DataValueField(String); 9 Description 10 Gets or sets the field of the data source that provides the value of each list 11 item. 12 Use this property to specify the field that contains the value of each item in 13 a list control. 14 Enabled 15 EnableViewState 16 **Events** 17 Font 18 ForeColor 19 HasChildViewState 20 Height 21 ID 22

IsTrackingViewState

TrackViewState

Items

Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets the collection of items in the list control.

Use this property to get the properties of items in the list control. This property can be used to determine the selected items in the list control.

NamingContainer

Page

Parent

SelectedIndex

**TrackViewState** 

## Description

Gets or sets the lowest ordinal index of the selected items in the list.

Use this property to determine the index of the currently selected item in the list if the list control allows only one selection. If the list control supports multiple selections, use this property to determine the lowest index of the selected items.

SelectedItem

**TrackViewState** 

[C#] public virtual ListItem SelectedItem {get;}

[C++] public: property virtual ListItem\* get SelectedItem();

[VB] Overridable Public ReadOnly Property SelectedItem As ListItem

3

5

6

7

8

9

10

11

12

13

17

24

[JScript] public function get SelectedItem(): ListItem; Description Gets the selected item with the lowest index in the list control. If the list control allows only a single selection, use this property to get the individual properties of the selected item. If the list control allows multiple selections, use this property to get the properties of the lowest indexed item selected from the list control. Site Style TabIndex TagKey TagName 14 **ToolTip** 15 UniqueID 16 ViewState ViewStateIgnoresCase 18 Visible 19 Width 20 TrackViewState 21 22 23

TemplateSourceDirectory Description 1964

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Occurs when the selection on the list changes and is posted back to the server. This event is raised when the selection from the list changes and is posted back to the server. ClearSelection [C#] public virtual void ClearSelection(); [C++] public: virtual void ClearSelection(); [VB] Overridable Public Sub ClearSelection() [JScript] public function ClearSelection(); Description Clears out the list selection and sets the  ${\bf System. Web. UI. WebControls. List Item. Selected}\ \ {\bf property}\ \ {\bf of}\ \ {\bf all}\ \ {\bf items}\ \ {\bf to}\ \ {\bf false}.$ LoadViewState [C#] protected override void LoadViewState(object savedState); [C++] protected: void LoadViewState(Object\* savedState); [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object)  $[JScript]\ protected\ override\ function\ LoadViewState (savedState: Object);$ Description Load previously saved state. Overridden to restore selection. **OnDataBinding** 

1	
2	[C#] protected override void OnDataBinding(EventArgs e);
3	[C++] protected: void OnDataBinding(EventArgs* e);
4	[VB] Overrides Protected Sub OnDataBinding(ByVal e As EventArgs)
5	[JScript] protected override function OnDataBinding(e: EventArgs);
6	
7	Description
8	OnSelectedIndexChanged
9	
10	[C#] protected virtual void OnSelectedIndexChanged(EventArgs e);
11	[C++] protected: virtual void OnSelectedIndexChanged(EventArgs* e);
12	[VB] Overridable Protected Sub OnSelectedIndexChanged(ByVal e As
13	EventArgs)
14	[JScript] protected function OnSelectedIndexChanged(e: EventArgs);
15	
16	Description
17	Raises the
18	System.Web.UI.WebControls.ListControl.SelectedIndexChanged event.
19	Raising an event invokes the event handler through a delegate. For more
20	information, see . An System.EventArgs that contains the event data.
21	SaveViewState
22	
23	[C#] protected override object SaveViewState();
24	[C++] protected: Object* SaveViewState();
25	[VB] Overrides Protected Function SaveViewState() As Object

1	[JScript] protected override function SaveViewState(): Object;
2	
3	Description
4	TrackViewState
5	
6	[C#] protected override void TrackViewState();
7	[C++] protected: void TrackViewState();
8	[VB] Overrides Protected Sub TrackViewState()
9	[JScript] protected override function TrackViewState();
10	
11	Description
12	ListItem class (System.Web.UI.WebControls)
13	TrackViewState
14	
15	
16	Description
17	Represents a data item in a databound list control. This class cannot be
18	inherited.
19	A System.Web.UI.WebControls.ListItem control represents an individual
20	data item within a databound list control, such as a
21	System.Web.UI.WebControls.ListBox or a
22	System.Web.UI.WebControls.RadioButtonList control.
23	ListItem
24	Example Syntax:
25	TrackViewState

1	
2	[C#] public ListItem();
3	[C++] public: ListItem();
4	[VB] Public Sub New()
5	[JScript] public function ListItem(); Initializes a new instance of the
6	System.Web.UI.WebControls.ListItem class.
7	
8	Description
9	Initializes a new instance of the System.Web.UI.WebControls.ListItem
10	class.
11	ListItem
12	Example Syntax:
13	TrackViewState
14	
15	[C#] public ListItem(string text);
16	[C++] public: ListItem(String* text);
17	[VB] Public Sub New(ByVal text As String)
18	[JScript] public function ListItem(text : String);
19	
20	Description
21	Initializes a new instance of the System.Web.UI.WebControls.ListItem
22	class with the specified text data. The text data to initialize the list item with.
23	ListItem
24	Example Syntax:
25	TrackViewState

[C#] public ListItem(string text, string value); 2 [C++] public: ListItem(String\* text, String\* value); 3 [VB] Public Sub New(ByVal text As String, ByVal value As String) [JScript] public function ListItem(text : String, value : String); 5 6 Description Initializes a new instance of the System.Web.UI.WebControls.ListItem 8 class with the specified text and value data. 9 The following table shows initial property values for an instance of 10 System.Web.UI.WebControls.ListItem . The text data to initialize the list item 11 with. The value data to initialize the list item with. 12 Attributes 13 **TrackViewState** 14 15 [C#] public AttributeCollection Attributes {get;} 16 [C++] public: property AttributeCollection\* get Attributes(); 17 [VB] Public ReadOnly Property Attributes As AttributeCollection 18 [JScript] public function get Attributes(): AttributeCollection; 19 20 Description 21 Gets the collection of attribute name/value pairs expressed on the list item 22 control but not supported by the control's strongly typed properties. 23 Selected 24 **TrackViewState** 

```
[C#] public bool Selected {get; set;}
    [C++] public: property bool get Selected();public: property void
    set Selected(bool);
    [VB] Public Property Selected As Boolean
5
    [JScript] public function get Selected(): Boolean; public function set
    Selected(Boolean);
7
8
    Description
           Specifies a value indicating whether the item is selected.
10
           Text
11
           TrackViewState
12
13
    [C#] public string Text {get; set;}
14
    [C++] public: property String* get Text(); public: property void
15
    set Text(String*);
16
    [VB] Public Property Text As String
17
    [JScript] public function get Text(): String; public function set Text(String);
18
19
    Description
20
           Gets or sets the text displayed in the list control for the item represented by
21
    the \ System. Web. UI. Web Controls. List Item \ control.
22
           If the list item control has no text content, then gets its value content, if
23
    exists, instead.
24
            Value
25
```

1	TrackViewState
2	
3	[C#] public string Value {get; set;}
4	[C++] public:property String* get_Value();public:property void
5	set_Value(String*);
6	[VB] Public Property Value As String
7	[JScript] public function get Value() : String; public function set Value(String);
8	
9	Description
10	Gets or sets the value content of the list item control.
11	If the list item control has no value content, then gets its text content, if
12	exists, instead.
13	Equals
14	
15	[C#] public override bool Equals(object o);
16	[C++] public: bool Equals(Object* 0);
17	[VB] Overrides Public Function Equals(ByVal o As Object) As Boolean
18	[JScript] public override function Equals(o : Object) : Boolean;
19	
20	Description
21	FromString
22	
23	[C#] public static ListItem FromString(string s);
24	[C++] public: static ListItem* FromString(String* s);
25	[VB] Public Shared Function FromString(ByVal s As String) As ListItem

[JScript] public static function FromString(s: String): ListItem; 2 Description 3 Creates a System.Web.UI.WebControls.ListItem from the specified string. The specified string for creating a 5 System.Web.UI.WebControls.ListItem. 6 GetHashCode 7 8 [C#] public override int GetHashCode(); 9 [C++] public: int GetHashCode(); 10 [VB] Overrides Public Function GetHashCode() As Integer 11 [JScript] public override function GetHashCode(): int; 12 13 Description 14 15 IAttributeAccessor.GetAttribute 16 17 [C#] string IAttributeAccessor.GetAttribute(string name); 18 [C++] String\* IAttributeAccessor::GetAttribute(String\* name); 19 [VB] Function GetAttribute(ByVal name As String) As String Implements 20 IAttributeAccessor.GetAttribute 21 [JScript] function IAttributeAccessor.GetAttribute(name : String) : String; 22 IAttributeAccessor.SetAttribute 23 24 [C#] void IAttributeAccessor.SetAttribute(string name, string value);

```
[C++] void IAttributeAccessor::SetAttribute(String* name, String* value);
1
    [VB] Sub SetAttribute(ByVal name As String, ByVal value As String)
2
    Implements IAttributeAccessor.SetAttribute
3
    [JScript] function IAttributeAccessor.SetAttribute(name : String, value : String);
4
           IParserAccessor.AddParsedSubObject
5
6
    [C#] void IParserAccessor.AddParsedSubObject(object obj);
7
    [C++] void IParserAccessor::AddParsedSubObject(Object* obj);
8
    [VB] Sub AddParsedSubObject(ByVal obj As Object) Implements
9
    IParserAccessor.AddParsedSubObject
10
    [JScript] function IParserAccessor.AddParsedSubObject(obj : Object);
11
           IStateManager.LoadViewState
12
13
    [C#] void IStateManager.LoadViewState(object state);
14
    [C++] void IStateManager::LoadViewState(Object* state);
15
    [VB] Sub LoadViewState(ByVal state As Object) Implements
16
    IStateManager.LoadViewState
17
    [JScript] function IStateManager.LoadViewState(state: Object);
18
           IStateManager.SaveViewState
19
20
    [C#] object IStateManager.SaveViewState();
21
    [C++] Object* IStateManager::SaveViewState();
22
    [VB] Function SaveViewState() As Object Implements
23
    IStateManager.SaveViewState
24
    [JScript] function IStateManager.SaveViewState(): Object;
25
```

1	IStateManager.TrackViewState
2	
3	[C#] void IStateManager.TrackViewState();
4	[C++] void IStateManager::TrackViewState();
5	[VB] Sub TrackViewState() Implements IStateManager.TrackViewState
6	[JScript] function IStateManager.TrackViewState();
7	ToString
8	
9	[C#] public override string ToString();
10	[C++] public: String* ToString();
11	[VB] Overrides Public Function ToString() As String
12	[JScript] public override function ToString() : String;
13	
14	Description
15	ListItemCollection class (System.Web.UI.WebControls)
16	ToString
17	
18	
19	Description
20	Encapsulates the System.Web.UI.WebControls.ListItem controls within
21	a System.Web.UI.WebControls.ListControl. This class cannot be inherited.
22	ListItemCollection
23	Example Syntax:
24	ToString
25	

```
[C#] public ListItemCollection();
    [C++] public: ListItemCollection();
    [VB] Public Sub New()
    [JScript] public function ListItemCollection();
6
    Description
           Initializes a new instance of the
8
    System. Web. UI. Web Controls. List I tem Collection \ class.
           Capacity
10
           ToString
11
12
    [C#] public int Capacity {get; set;}
13
    [C++] public: __property int get_Capacity();public: __property void
14
    set_Capacity(int);
15
    [VB] Public Property Capacity As Integer
16
    [JScript] public function get Capacity(): int;public function set Capacity(int);
17
           Count
18
           ToString
19
20
    [C#] public int Count {get;}
21
    [C++] public: _ property int get Count();
22
    [VB] Public ReadOnly Property Count As Integer
23
    [JScript] public function get Count(): int;
24
25
```

```
Description
2
           Gets the item count of the collection.
3
           IsReadOnly
           ToString
5
6
    [C#] public bool IsReadOnly {get;}
    [C++] public: property bool get IsReadOnly();
8
    [VB] Public ReadOnly Property IsReadOnly As Boolean
    [JScript] public function get IsReadOnly(): Boolean;
10
11
    Description
12
           Gets a value indicating whether the collection is read-only.
13
           IsSynchronized
14
           ToString
15
16
    [C#] public bool IsSynchronized {get;}
17
    [C++] public: property bool get IsSynchronized();
18
    [VB] Public ReadOnly Property IsSynchronized As Boolean
19
    [JScript] public function get IsSynchronized(): Boolean;
20
21
    Description
22
           Gets a value indicating whether access to the collection is synchronized
23
    (thread-safe).
24
           Item
25 |
```

1	ToString
2	
3	[C#] public ListItem this[int index] {get;}
4	[C++] public:property ListItem* get_Item(int index);
5	[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As
6	ListItem
7	[JScript] returnValue = ListItemCollectionObject.Item(index);
8	
9	Description
10	Gets a System.Web.UI.WebControls.ListItem referenced by the specified
11	ordinal index value. An ordinal index value that specifies which
12	System.Web.UI.WebControls.ListItem to return.
13	SyncRoot
14	ToString
15	
16	[C#] public object SyncRoot {get;}
17	[C++] public:property Object* get_SyncRoot();
18	[VB] Public ReadOnly Property SyncRoot As Object
19	[JScript] public function get SyncRoot(): Object;
20	
21	Description
22	Gets the object that can be used to synchronize access to the collection. In
23	this case, it is the collection itself.
24	Add
25	

[C#] public void Add(ListItem item); [C++] public: void Add(ListItem\* item); 3 [VB] Public Sub Add(ByVal item As ListItem) [JScript] public function Add(item: ListItem); 6 Description Adds the specified System. Web. UI. Web Controls. List I tem to the end of 8 the collection. The System.Web.UI.WebControls.ListItem to add to the collection. 10 Add 11 12 [C#] public void Add(string item); 13 [C++] public: void Add(String\* item); 14 [VB] Public Sub Add(ByVal item As String) 15 [JScript] public function Add(item: String); Adds the specified item to the end of the collection. 17 18 Description 19 Adds the specified item to the end of the collection. A System.String that 20 specifies the item to add. 21 AddRange 22 23 [C#] public void AddRange(ListItem[] items); 24 [C++] public: void AddRange(ListItem\* items[]);

```
[VB] Public Sub AddRange(ByVal items() As ListItem)
    [JScript] public function AddRange(items : ListItem[]);
           Clear
3
    [C#] public void Clear();
    [C++] public: sealed void Clear();
    [VB] NotOverridable Public Sub Clear()
    [JScript] public function Clear();
9
    Description
10
           Removes all System. Web.UI. WebControls. ListItem controls from the
11
    collection.
           Contains
13
14
    [C#] public bool Contains(ListItem item);
15
    [C++] public: bool Contains(ListItem* item);
16
    [VB] Public Function Contains(ByVal item As ListItem) As Boolean
    [JScript] public function Contains(item : ListItem) : Boolean;
18
19
    Description
20
           Returns a value indicating whether the collection contains the specified
21
    item.
22
    Return Value: true if the collection contains the specified item; otherwise, false.
23
    A System. Web. UI. WebControls. ListItem to search for in the collection.
24
           CopyTo
```

```
[C#] public void CopyTo(Array array, int index);
2
    [C++] public: sealed void CopyTo(Array* array, int index);
3
    [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
    Integer)
    [JScript] public function CopyTo(array : Array, index : int);
7
    Description
8
           Copies contents from the collection to a specified System. Array with a
    specified starting index. The specified System.Array that receives copied contents
10
    from the collection. The starting position in the specified System. Array to receive
11
    copied contents.
12
           FindByText
13
14
    [C#] public ListItem FindByText(string text);
15
    [C++] public: ListItem* FindByText(String* text);
16
    [VB] Public Function FindByText(ByVal text As String) As ListItem
17
    [JScript] public function FindByText(text : String) : ListItem;
18
           FindByValue
19
20
    [C#] public ListItem FindByValue(string value);
21
    [C++] public: ListItem* FindByValue(String* value);
22
    [VB] Public Function FindByValue(ByVal value As String) As ListItem
23
    [JScript] public function FindByValue(value : String) : ListItem;
24
           GetEnumerator
25
```

1	
2	[C#] public IEnumerator GetEnumerator();
3	[C++] public:sealed IEnumerator* GetEnumerator();
4	[VB] NotOverridable Public Function GetEnumerator() As IEnumerator
5	[JScript] public function GetEnumerator(): IEnumerator;
6	
7	Description
8	Returns an enumerator of all System.Web.UI.WebControls.ListItem
9	controls within the collection.
10	Return Value: An enumerator that enumerates over all
11	System.Web.UI.WebControls.ListItem controls within the collection.
12	IndexOf
13	
14	[C#] public int IndexOf(ListItem item);
15	[C++] public: int IndexOf(ListItem* item);
16	[VB] Public Function IndexOf(ByVal item As ListItem) As Integer
17	[JScript] public function IndexOf(item : ListItem) : int;
18	
19	Description
20	Returns an ordinal index value that represents the position of the specified
21	System.Web.UI.WebControls.ListItem within the collection.
22	Return Value: The ordinal index position of the specified
23	System.Web.UI.WebControls.ListItem within the collection. The specified
24	System.Web.UI.WebControls.ListItem to search for in the collection.
25	Insert

25

1 [C#] public void Insert(int index, ListItem item); [C++] public: void Insert(int index, ListItem\* item); [VB] Public Sub Insert(ByVal index As Integer, ByVal item As ListItem) [JScript] public function Insert(index : int, item : ListItem); 6 Description 7 Inserts the specified System. Web. UI. Web Controls. ListItem to the 8 collection at the specified index location. The location in the collection to add the 9 System.Web.UI.WebControls.ListItem. The 10 System.Web.UI.WebControls.ListItem to add to the collection. 11 Insert 12 13 [C#] public void Insert(int index, string item); 14 [C++] public: void Insert(int index, String\* item); 15 [VB] Public Sub Insert(ByVal index As Integer, ByVal item As String) 16 [JScript] public function Insert(index: int, item: String); Inserts the specified item 17 to the collection at the specified index location. 18 19 Description 20 Adds the specified item to the collection at the specified index location. 21 The location in the collection to add the System. Web.UI. WebControls. ListItem. 22 The item to add to the collection. 23 Remove

[C#] public void Remove(ListItem item); 2 [C++] public: void Remove(ListItem\* item); 3 [VB] Public Sub Remove(ByVal item As ListItem) [JScript] public function Remove(item: ListItem); 6 Description Removes the specified System. Web. UI. Web Controls. ListItem from the 8 collection. The System. Web.UI. WebControls. ListItem to remove from the collection. 10 Remove 11 12 [C#] public void Remove(string item); 13 [C++] public: void Remove(String\* item); 14 [VB] Public Sub Remove(ByVal item As String) 15 [JScript] public function Remove(item : String); Removes the specified item from 16 the collection. 17 18 Description 19 Removes the specified item from the collection. The item to remove from 20 the collection. 21 RemoveAt 22 23 [C#] public void RemoveAt(int index); 24 [C++] public: \_\_sealed void RemoveAt(int index);

1	[VB] NotOverridable Public Sub RemoveAt(ByVal index As Integer)
2	[JScript] public function RemoveAt(index : int);
3	
4	Description
5	Removes the System.Web.UI.WebControls.ListItem from the collection
6	at the specified index location. The location in the collection to remove the
7	System.Web.UI.WebControls.ListItem.
8	IList.Add
9	
10	[C#] int IList.Add(object item);
11	[C++] int IList::Add(Object* item);
12	[VB] Function Add(ByVal item As Object) As Integer Implements IList.Add
13	[JScript] function IList.Add(item : Object) : int;
14	IList.Contains
15	
16	[C#] bool IList.Contains(object item);
17	[C++] bool IList::Contains(Object* item);
18	[VB] Function Contains(ByVal item As Object) As Boolean Implements
19	IList.Contains
20	[JScript] function IList.Contains(item : Object) : Boolean;
21	IList.IndexOf
22	
23	[C#] int IList.IndexOf(object item);
24	[C++] int IList::IndexOf(Object* item);
25	[VB] Function IndexOf(ByVal item As Object) As Integer Implements

```
IList.IndexOf
    [JScript] function IList.IndexOf(item: Object): int;
2
           IList.Insert
3
    [C#] void IList.Insert(int index, object item);
    [C++] void IList::Insert(int index, Object* item);
    [VB] Sub Insert(ByVal index As Integer, ByVal item As Object) Implements
    IList.Insert
    [JScript] function IList.Insert(index : int, item : Object);
           IList.Remove
10
11
    [C#] void IList.Remove(object item);
12
    [C++] void IList::Remove(Object* item);
13
    [VB] Sub Remove(ByVal item As Object) Implements IList.Remove
    [JScript] function IList.Remove(item : Object);
15
           IStateManager.LoadViewState
16
17
    [C#] void IStateManager.LoadViewState(object state);
    [C++] void IStateManager::LoadViewState(Object* state);
19
    [VB] Sub LoadViewState(ByVal state As Object) Implements
20
    IStateManager.LoadViewState
21
    [JScript] function IStateManager.LoadViewState(state : Object);
22
           IStateManager.SaveViewState
23
24
    [C#] object IStateManager.SaveViewState();
```

```
[C++] Object* IStateManager::SaveViewState();
    [VB] Function SaveViewState() As Object Implements
    IStateManager.SaveViewState
    [JScript] function IStateManager.SaveViewState(): Object;
           IStateManager.TrackViewState
5
6
    [C#] void IStateManager.TrackViewState();
    [C++] void IStateManager::TrackViewState();
    [VB] Sub TrackViewState() Implements IStateManager.TrackViewState
    [JScript] function IStateManager.TrackViewState();
10
          ListItemControlBuilder class (System.Web.UI.WebControls)
11
           ToString
12
13
14
    Description
15
           Interacts with the parser to build a
16
    System.Web.UI.WebControls.ListItem control.
           ListItemControlBuilder
18
          Example Syntax:
19
           ToString
20
21
    [C#] public ListItemControlBuilder();
22
    [C++] public: ListItemControlBuilder();
23
    [VB] Public Sub New()
    [JScript] public function ListItemControlBuilder();
```

1	ControlType
2	FChildrenAsProperties
3	FIsNonParserAccessor
4	HasAspCode
5	ID
6	InDesigner
7	NamingContainerType
8	Parser
9	TagName
10	AllowWhitespaceLiterals
11	
12	[C#] public override bool AllowWhitespaceLiterals();
13	[C++] public: bool AllowWhitespaceLiterals();
14	[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
15	[JScript] public override function AllowWhitespaceLiterals(): Boolean;
16	
17	Description
18	
19	HtmlDecodeLiterals
20	
21	[C#] public override bool HtmlDecodeLiterals();
22	[C++] public: bool HtmlDecodeLiterals();
23	[VB] Overrides Public Function HtmlDecodeLiterals() As Boolean
24	[JScript] public override function HtmlDecodeLiterals(): Boolean;
25	

ToString

1	
2	Description
3	
4	ListItemType enumeration (System.Web.UI.WebControls)
5	ToString
6	
7	
8	Description
9	Specifies the type of an item in a list control.
10	The System.Web.UI.WebControls.ListItemType enumeration represents
11	the different items that can be included in a list control, such as
12	System.Web.UI.WebControls.DataGrid ,
13	System.Web.UI.WebControls.DataList, and the
14	System.Web.UI.WebControls.Repeater . A typical list control consists of cells
15	that contain elements represented by this enumeration.
16	ToString
17	
18	[C#] public const ListItemType AlternatingItem;
19	[C++] public: const ListItemType AlternatingItem;
20	[VB] Public Const AlternatingItem As ListItemType
21	[JScript] public var AlternatingItem : ListItemType;
22	
23	Description
24	An item in alternating (zero-based even-indexed) cells. It is databound.

1	
1	
2	[C#] public const ListItemType EditItem;
3	[C++] public: const ListItemType EditItem;
4	[VB] Public Const EditItem As ListItemType
5	[JScript] public var EditItem : ListItemType;
6	
7	Description
8	An item in a list control currently in edit mode. It is databound.
9	ToString
10	
11	[C#] public const ListItemType Footer;
12	[C++] public: const ListItemType Footer;
13	[VB] Public Const Footer As ListItemType
14	[JScript] public var Footer: ListItemType;
15	
16	Description
17	A footer for the list control. It is not databound.
18	ToString
19	
20	[C#] public const ListItemType Header;
21	[C++] public: const ListItemType Header;
22	[VB] Public Const Header As ListItemType
23	[JScript] public var Header: ListItemType;
24	
25	Description

1	A header for the list control. It is not databound.
2	ToString
3	
4	[C#] public const ListItemType Item;
5	[C++] public: const ListItemType Item;
6	[VB] Public Const Item As ListItemType
7	[JScript] public var Item: ListItemType;
8	
9	Description
10	An item in the list control. It is databound.
11	ToString
12	
13	[C#] public const ListItemType Pager;
14	[C++] public: const ListItemType Pager;
15	[VB] Public Const Pager As ListItemType
16	[JScript] public var Pager: ListItemType;
17	
18	Description
19	A pager that displays the controls to navigate to different pages associated
20	with the System.Web.UI.WebControls.DataGrid control. It is not databound.
21	ToString
22	
23	[C#] public const ListItemType SelectedItem;
24	[C++] public: const ListItemType SelectedItem;
25	[VB] Public Const SelectedItem As ListItemType

[JScript] public var SelectedItem : ListItemType; 2 Description A selected item in the list control. It is databound. **ToString** 5 6 [C#] public const ListItemType Separator; 7 [C++] public: const ListItemType Separator; [VB] Public Const Separator As ListItemType [JScript] public var Separator : ListItemType; 10 11 Description 12 A separator between items in a list control. It is not databound. 13 ListSelectionMode enumeration (System.Web.UI.WebControls) 14 **ToString** 15 16 17 Description 18 Specifies the selection mode of the System.Web.UI.WebControls.ListBox 19 control. 20 The System.Web.UI.WebControls.ListSelectionMode enumeration 21 represents the selection mode of the System.Web.UI.WebControls.ListBox 22 control that determines whether a user can select multiple items or just a single 23 item. 24 **ToString** 

1	
2	[C#] public const ListSelectionMode Multiple;
3	[C++] public: const ListSelectionMode Multiple;
4	[VB] Public Const Multiple As ListSelectionMode
5	[JScript] public var Multiple : ListSelectionMode;
6	
7	Description
8	Multiple item selection mode.
9	ToString
10	
11	[C#] public const ListSelectionMode Single;
12	[C++] public: const ListSelectionMode Single;
13	[VB] Public Const Single As ListSelectionMode
14	[JScript] public var Single : ListSelectionMode;
15	
16	Description
17	Single item selection mode.
18	Literal class (System.Web.UI.WebControls)
19	ToString
20	
21	
22	Description
23	
24	Literal
25	Example Syntax:

ViewStateIgnoresCase

24

25

1993 MS1-863US.APP lee@hayes plic 509-324-9256

1	Visible
2	AddParsedSubObject
3	
4	[C#] protected override void AddParsedSubObject(object obj);
5	[C++] protected: void AddParsedSubObject(Object* obj);
6	[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)
7	[JScript] protected override function AddParsedSubObject(obj : Object);
8	CreateControlCollection
9	
10	[C#] protected override ControlCollection CreateControlCollection();
11	[C++] protected: ControlCollection* CreateControlCollection();
12	[VB] Overrides Protected Function CreateControlCollection() As
13	ControlCollection
14	[JScript] protected override function CreateControlCollection():
15	ControlCollection;
16	Render
17	
18	[C#] protected override void Render(HtmlTextWriter output);
19	[C++] protected: void Render(HtmlTextWriter* output);
20	[VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)
21	[JScript] protected override function Render(output : HtmlTextWriter);
22	LiteralControlBuilder class (System.Web.UI.WebControls)
23	TrackViewState
24	LiteralControlBuilder
25	Example Syntax:

1	TrackViewState
2	ControlType
3	FChildrenAsProperties
4	FIsNonParserAccessor
5	HasAspCode
6	ID
7	InDesigner
8	NamingContainerType
9	Parser
10	TagName
11	AllowWhitespaceLiterals
12	
13	[C#] public override bool AllowWhitespaceLiterals();
14	[C++] public: bool AllowWhitespaceLiterals();
15	[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
16	[JScript] public override function AllowWhitespaceLiterals(): Boolean;
17	AppendSubBuilder
18	
19	[C#] public override void AppendSubBuilder(ControlBuilder subBuilder)
20	[C++] public: void AppendSubBuilder(ControlBuilder* subBuilder);
21	[VB] Overrides Public Sub AppendSubBuilder(ByVal subBuilder As
22	ControlBuilder)
23	[JScript] public override function AppendSubBuilder(subBuilder:
24	ControlBuilder);
25	MonthChangedEventArgs class (System.Web.UI.WebControls)

1	ToString
2	
3	
4	Description
5	Provides data for the
6	System.Web.UI.WebControls.Calendar.VisibleMonthChanged event of a
7	System.Web.UI.WebControls.Calendar. This class cannot be inherited.
8	The System.Web.UI.WebControls.Calendar.VisibleMonthChanged
9	event is raised when the user clicks on the
10	System.Web.UI.WebControls.Calendar navigation controls to display the next
11	or previous month.
12	MonthChangedEventArgs
13	Example Syntax:
14	ToString
15	
16	[C#] public MonthChangedEventArgs(DateTime newDate, DateTime
17	previousDate);
18	[C++] public: MonthChangedEventArgs(DateTime newDate, DateTime
19	previousDate);
20	[VB] Public Sub New(ByVal newDate As DateTime, ByVal previousDate As
21	DateTime)
22	[JScript] public function MonthChangedEventArgs(newDate : DateTime,
23	previousDate : DateTime);
24	
25	Description

· ·

1

3

7

8

9

10

11

12

13

15

16

17

18

19

20

21

22

23

24

Initializes a new instance of the

System.Web.UI.WebControls.MonthChangedEventArgs class.

The following table shows initial property values for an instance of System.Web.UI.WebControls.MonthChangedEventArgs. The date that determines the month currently displayed by the

System.Web.UI.WebControls.Calendar. The date that determines the month previously displayed by the System.Web.UI.WebControls.Calendar.

NewDate

**ToString** 

[C#] public DateTime NewDate {get;}

[C++] public: property DateTime get NewDate();

[VB] Public ReadOnly Property NewDate As DateTime

[JScript] public function get NewDate(): DateTime;

Description

Gets the date that determines the currently displayed month in the System.Web.UI.WebControls.Calendar.

Use this property to get the month currently displayed by the **System.Web.UI.WebControls.Calendar** and to compare the currently displayed month with another month. For example, you can use this property to calculate the number of months until or since a certain date.

**PreviousDate** 

**ToString** 

7

[C#] public DateTime PreviousDate {get;} [C++] public: \_property DateTime get PreviousDate(); [VB] Public ReadOnly Property PreviousDate As DateTime [JScript] public function get PreviousDate(): DateTime;

## Description

Gets the date that determined the previously displayed month in the System.Web.UI.WebControls.Calendar.

Use this property to determine the previously displayed month on the System. Web. UI. Web Controls. Calendar. This date can be used to restore the System.Web.UI.WebControls.Calendar back to the previously displayed month by setting the System.Web.UI.WebControls.Calendar.VisibleDate property to  $System. Web. UI. Web Controls. Month Change d Event Args. Previous Date \ .$ 

MonthChangedEventHandler delegate (System.Web.UI.WebControls) **ToString** 

## Description

Represents the method that handles the System.Web.UI.WebControls.Calendar.VisibleMonthChanged event of a System.Web.UI.WebControls.Calendar. The source of the event. A System.Web.UI.WebControls.MonthChangedEventArgs that contains the event data.

25

24

18

19

20

21

22

23

3

5

6

9

10

11

12

13

14

15

17

18

19

20

21

22

23

24

25

The System.Web.UI.WebControls.Calendar.VisibleMonthChanged event is raised when the user clicks on the navigation controls for the System. Web. UI. Web Controls. Calendar to display the next or previous month. NextPrevFormat enumeration (System.Web.UI.WebControls) **ToString** Description Represents the display format for the previous and next month navigation controls within the System.Web.UI.WebControls.Calendar. The System.Web.UI.WebControls.NextPrevFormat enumeration represents the different styles for the next and previous month buttons on the System.Web.UI.WebControls.Calendar. **ToString** [C#] public const NextPrevFormat CustomText; [C++] public: const NextPrevFormat CustomText; [VB] Public Const CustomText As NextPrevFormat [JScript] public var CustomText : NextPrevFormat; Description Custom text format for the month navigation controls on the

System.Web.UI.WebControls.Calendar.

**ToString** 

1	
2	[C#] public const NextPrevFormat FullMonth;
3	[C++] public: const NextPrevFormat FullMonth;
4	[VB] Public Const FullMonth As NextPrevFormat
5	[JScript] public var FullMonth : NextPrevFormat;
6	
7	Description
8	Full month name format for the month navigation controls on the
9	System.Web.UI.WebControls.Calendar. For example, "January".
10	ToString
11	
12	[C#] public const NextPrevFormat ShortMonth;
13	[C++] public: const NextPrevFormat ShortMonth;
14	[VB] Public Const ShortMonth As NextPrevFormat
15	[JScript] public var ShortMonth : NextPrevFormat;
16	
17	Description
18	Abbreviated month name format for the month navigation controls on the
19	System.Web.UI.WebControls.Calendar. For example, "Jan".
20	PagedDataSource class (System.Web.UI.WebControls)
21	ToString
22	
23	
24	Description
25	

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Provides a wrapper over an **System.Collections.ICollection** data source to implement paging semantics or 'paged views' on top of the underlying datasource. This class cannot be inherited.

This class uses the best available interface to enumerate over the data belonging to the current page. If the underlying data source supports indexed access (like System.Array and System.Collections.IList), this wrapper uses it. Otherwise it falls back on System.Collections.IEnumerable.

PagedDataSource

Example Syntax:

**ToString** 

[C#] public PagedDataSource();

[C++] public: PagedDataSource();

[VB] Public Sub New()

[JScript] public function PagedDataSource();

Description

Initializes a new instance of the

System.Web.UI.WebControls.PagedDataSource class.

The following table shows initial property values for an instance of System.Web.UI.WebControls.PagedDataSource.

AllowCustomPaging

**ToString** 

[C#] public bool AllowCustomPaging {get; set;}

```
[C++] public: property bool get AllowCustomPaging();public: property void
    set AllowCustomPaging(bool);
2
    [VB] Public Property Allow Custom Paging As Boolean
    [JScript] public function get AllowCustomPaging(): Boolean; public function set
    AllowCustomPaging(Boolean);
 6
    Description
           Gets or sets a value indicating whether to assume the underlying data
 8
    source contains data for just the current page.
 9
           This is only applicable if AllowPaging is true. In this scenario, the
10
    developer can fetch just the right number of items from the DataSet to fill up a
11
    single page, which may be more efficient based on the source of the data.
12
           AllowPaging
13
           ToString
14
15
    [C#] public bool AllowPaging {get; set;}
16
    [C++] public: property bool get AllowPaging(); public: property void
17
    set_AllowPaging(bool);
18
    [VB] Public Property AllowPaging As Boolean
19
    [JScript] public function get AllowPaging(): Boolean; public function set
20
    AllowPaging(Boolean);
21
22
    Description
23
           Gets or sets a value indicating whether to implement page semantics on top
24
    of the underlying datasource.
25
```

	11	
	1	Count
	2	ToString
	3	
	4	[C#] public int Count {get;}
	5	[C++] public:property int get_Count();
	6	[VB] Public ReadOnly Property Count As Integer
	7	[JScript] public function get Count(): int;
	8	
	9	Description
	10	Gets the number of items to be used from the datasource.
desert the state of the state of	11	This takes into account several factors such as paged/non-paged modes,
, mm, , mm,	12	custom paging, and last page.
	13	CurrentPageIndex
	14	ToString
	15	
	16	[C#] public int CurrentPageIndex {get; set;}
	17	[C++] public:property int get_CurrentPageIndex();public:property void
	18	set_CurrentPageIndex(int);
	19	[VB] Public Property CurrentPageIndex As Integer
	20	[JScript] public function get CurrentPageIndex(): int;public function set
	21	CurrentPageIndex(int);
	22	
	23	Description
	24	Gets or sets a value indicating the index of the current page.
	25	DataSource

	1	ToString
	2	
	3	[C#] public IEnumerable DataSource {get; set;}
	4	[C++] public:property IEnumerable* get_DataSource();public:property void
	5	set_DataSource(IEnumerable*);
	6	[VB] Public Property DataSource As IEnumerable
	7	[JScript] public function get DataSource(): IEnumerable; public function set
	8	DataSource(IEnumerable);
	9	
And the state that	10	Description
	11	Gets or sets the data source.
	12	DataSourceCount
	13	ToString
	14	
	15	[C#] public int DataSourceCount {get;}
	16	[C++] public:property int get_DataSourceCount();
	17	[VB] Public ReadOnly Property DataSourceCount As Integer
	18	[JScript] public function get DataSourceCount(): int;
	19	
	20	Description
	21	Gets the item count of the data source.
	22	FirstIndexInPage
	23	ToString
	24	
	25	[C#] public int FirstIndexInPage {get;}

```
[C++] public: property int get FirstIndexInPage();
    [VB] Public ReadOnly Property FirstIndexInPage As Integer
2
    [JScript] public function get FirstIndexInPage(): int;
4
    Description
5
           Gets the first index in the page.
6
           IsCustomPagingEnabled
           ToString
8
9
    [C#] public bool IsCustomPagingEnabled {get;}
10
    [C++] public: property bool get IsCustomPagingEnabled();
11
    [VB] Public ReadOnly Property IsCustomPagingEnabled As Boolean
12
    [JScript] public function get IsCustomPagingEnabled(): Boolean;
13
14
    Description
15
           Gets a value indicating whether custom paging is enabled.
16
           IsFirstPage
17
           ToString
18
19
    [C#] public bool IsFirstPage {get;}
20
    [C++] public: property bool get IsFirstPage();
21
    [VB] Public ReadOnly Property IsFirstPage As Boolean
22
    [JScript] public function get IsFirstPage(): Boolean;
23
24
    Description
25
```

1	Gets a value indicating if the current page is the first page.
2	IsLastPage
3	ToString
4	
5	[C#] public bool IsLastPage {get;}
6	[C++] public:property bool get_IsLastPage();
7	[VB] Public ReadOnly Property IsLastPage As Boolean
8	[JScript] public function get IsLastPage(): Boolean;
9	
10	Description
11	Gets a value indicating if the current page is the last page.
12	IsPagingEnabled
13	ToString
14	
15	[C#] public bool IsPagingEnabled {get;}
16	[C++] public:property bool get_IsPagingEnabled();
17	[VB] Public ReadOnly Property IsPagingEnabled As Boolean
18	[JScript] public function get IsPagingEnabled(): Boolean;
19	
20	Description
21	Gets a value indicating whether paging is enabled.
22	IsReadOnly
23	ToString
24	
25	[C#] public bool IsReadOnly {get·}

```
[C++] public: property bool get IsReadOnly();
    [VB] Public ReadOnly Property IsReadOnly As Boolean
    [JScript] public function get IsReadOnly(): Boolean;
    Description
 5
           Gets a value indicating whether the data source is read-only.
           IsSynchronized
           ToString
8
9
    [C#] public bool IsSynchronized {get;}
10
    [C++] public: property bool get IsSynchronized();
11
    [VB] Public ReadOnly Property IsSynchronized As Boolean
12
    [JScript] public function get IsSynchronized(): Boolean;
13
14
    Description
15
           Gets a value indicating whether access to the data source is synchronized
16
    (thread-safe).
17
           PageCount
18
           ToString
19
20
    [C#] public int PageCount {get;}
21
    [C++] public: property int get PageCount();
22
    [VB] Public ReadOnly Property PageCount As Integer
23
    [JScript] public function get PageCount(): int;
24
25
```

```
Description
           Gets the page count.
3
           PageSize
           ToString
    [C#] public int PageSize {get; set;}
    [C++] public: property int get PageSize(); public: property void
8
    set_PageSize(int);
9
    [VB] Public Property PageSize As Integer
10
    [JScript] public function get PageSize(): int;public function set PageSize(int);
11
12
    Description
13
           Gets or sets the page size.
14
           SyncRoot
15
           ToString
16
17
    [C#] public object SyncRoot {get;}
18
    [C++] public: property Object* get SyncRoot();
19
    [VB] Public ReadOnly Property SyncRoot As Object
20
    [JScript] public function get SyncRoot(): Object;
21
22
    Description
23
           Gets the object that can be used to synchronize access to the collection. In
24
    this case, it is the collection itself.
25
```

1	VirtualCount
2	ToString
3	
4	[C#] public int VirtualCount {get; set;}
5	[C++] public:property int get_VirtualCount();public:property void
6	set_VirtualCount(int);
7	[VB] Public Property VirtualCount As Integer
8	[JScript] public function get VirtualCount(): int;public function set
9	VirtualCount(int);
10	
11	Description
12	Gets or sets the virtual count.
13	СоруТо
14	
15	[C#] public void CopyTo(Array array, int index);
16	[C++] public:sealed void CopyTo(Array* array, int index);
17	[VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
18	Integer)
19	[JScript] public function CopyTo(array: Array, index: int);
20	
21	Description
22	Copies contents from the data source to the specified System.Array with
23	the specified starting index. The specified System.Array that receives copied
24	contents from the data source. The starting position in the specified System.Array
25	to receive copied contents.

1	GetEnumerator
2	
3	[C#] public IEnumerator GetEnumerator();
4	[C++] public:sealed IEnumerator* GetEnumerator();
5	[VB] NotOverridable Public Function GetEnumerator() As IEnumerator
6	[JScript] public function GetEnumerator(): IEnumerator;
7	
8	Description
9	Returns an enumerator of all items within the data source.
10	Return Value: An enumerator that enumerates over all items within the data
11	source.
12	GetItemProperties
13	
14	[C#] public PropertyDescriptorCollection GetItemProperties(PropertyDescriptor[]
15	listAccessors);
16	[C++] public:sealed PropertyDescriptorCollection*
17	GetItemProperties(PropertyDescriptor* listAccessors[]);
18	[VB] NotOverridable Public Function GetItemProperties(ByVal listAccessors()
19	As PropertyDescriptor) As PropertyDescriptorCollection
20	[JScript] public function GetItemProperties(listAccessors : PropertyDescriptor[]) :
21	PropertyDescriptorCollection;
22	GetListName
23	
24	[C#] public string GetListName(PropertyDescriptor[] listAccessors);
25	[C++] public:sealed String* GetListName(PropertyDescriptor*

1	listAccessors[]);
2	[VB] NotOverridable Public Function GetListName(ByVal listAccessors() As
3	PropertyDescriptor) As String
4	[JScript] public function GetListName(listAccessors : PropertyDescriptor[]) :
5	String;
6	PagerMode enumeration (System.Web.UI.WebControls)
7	ToString
8	
9	
10	Description
11	Represents the mode of the pager for accessing various pages within the
12	System.Web.UI.WebControls.DataGrid control.
13	The System.Web.UI.WebControls.PagerMode enumeration represents
14	the different display modes for the pager of a
15	System.Web.UI.WebControls.DataGrid control.
16	ToString
17	
18	[C#] public const PagerMode NextPrev;
19	[C++] public: const PagerMode NextPrev;
20	[VB] Public Const NextPrev As PagerMode
21	[JScript] public var NextPrev : PagerMode;
22	
23	Description
24	A pager with <b>Previous</b> and <b>Next</b> buttons to access the next and previous
25	pages.

1	ToString
2	
3	[C#] public const PagerMode NumericPages;
4	[C++] public: const PagerMode NumericPages;
5	[VB] Public Const NumericPages As PagerMode
6	[JScript] public var NumericPages : PagerMode;
7	
8	Description
9	A pager with numbered buttons to access pages directly.
10	PagerPosition enumeration (System.Web.UI.WebControls)
11	ToString
12	
13	
14	Description
15	Specifies the position of the pager for accessing various pages within the
16	System.Web.UI.WebControls.DataGrid control.
17	The System.Web.UI.WebControls.PagerPosition enumeration represents
18	the different locations where the pager can be displayed.
19	ToString
20	
21	[C#] public const PagerPosition Bottom;
22	[C++] public: const PagerPosition Bottom;
23	[VB] Public Const Bottom As PagerPosition
24	[JScript] public var Bottom: PagerPosition;

Description 2 A pager positioned at the bottom of the 3 System.Web.UI.WebControls.DataGrid . **ToString** 5 6 [C#] public const PagerPosition Top; [C++] public: const PagerPosition Top; 8 [VB] Public Const Top As PagerPosition 9 [JScript] public var Top: PagerPosition; 10 11 Description 12 A pager positioned at the top of the 13 System. Web. UI. Web Controls. Data Grid.14 **ToString** 15 16 [C#] public const PagerPosition TopAndBottom; 17 [C++] public: const PagerPosition TopAndBottom; 18 [VB] Public Const TopAndBottom As PagerPosition 19 [JScript] public var TopAndBottom: PagerPosition; 20 21 Description 22 Pagers positioned at both the top and the bottom of the 23 System.Web.UI.WebControls.DataGrid. 24 Panel class (System.Web.UI.WebControls) 25

1	ToString
2	
3	
4	Description
5	Represents a control that acts as a container for other controls.
6	The System.Web.UI.WebControls.Panel control is a container for other
7	controls. It is especially useful when you want to generate controls
8	programmatically or hide/show a group of controls.
9	Panel
10	Example Syntax:
11	ToString
12	
13	[C#] public Panel();
14	[C++] public: Panel();
15	[VB] Public Sub New()
16	[JScript] public function Panel();
17	
18	Description
19	Initializes a new instance of the System.Web.UI.WebControls.Panel
20	class.
21	AccessKey
22	Attributes
23	BackColor
24	BackImageUrl
25	ToString

1	
2	
3	Description
4	Gets or sets the URL of the background image for t
5	Use this property to display a custom image for the
6	System.Web.UI.WebControls.Panel control.
7	BorderColor
8	BorderStyle
9	BorderWidth
10	ChildControlsCreated
11	ClientID
12	Context
13	Controls
14	ControlStyle
15	ControlStyleCreated
16	CssClass
17	Enabled
18	EnableViewState
19	Events
20	Font
21	ForeColor
22	HasChildViewState
23	Height
24	HorizontalAlign
25	ToString
1.1	•

of the background image for the panel control.

1	
2	
3	Description
4	Gets or sets the horizontal alignment of the contents within the panel.
5	Use this property to specify the alignment of the elements in the
6	System.Web.UI.WebControls.Panel .
7	ID
8	IsTrackingViewState
9	NamingContainer
10	Page
11	Parent
12	Site
13	Style
14	TabIndex
15	TagKey
16	TagName
17	TemplateSourceDirectory
18	ToolTip
19	UniqueID
20	ViewState
21	ViewStateIgnoresCase
22	Visible
23	Width
24	Wrap
25	ToString

lee@hayes pilc 509-324-9256 2016 MS1-863US.APP

Description 3 Gets or sets a value indicating whether the content wraps within the panel. Use this property to wrap the contents of the System.Web.UI.WebControls.Panel. AddAttributesToRender 7 8 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer); 9 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer); 10 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As 11 HtmlTextWriter) 12 [JScript] protected override function AddAttributesToRender(writer: 13 HtmlTextWriter); 14 15 Description 16 Add background-image to list of style attributes to render. Add align and 17 nowrap to list of attributes to render. 18 PlaceHolder class (System.Web.UI.WebControls) 19 **TrackViewState** 20 21 22 Description 23 A container to store dynamically added server controls on the Web page. 24 25

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Page

Use the System. Web.UI. WebControls. PlaceHolder control as a container to store dynamically added server controls to the Web page. The System.Web.UI.WebControls.PlaceHolder control does not produce any visible output and is only used as a container for other controls on the Web page. You can use the System. Web. UI. Control. Controls collection to add, insert, or remove a control from the System.Web.UI.WebControls.PlaceHolder control. PlaceHolder Example Syntax: **TrackViewState** [C#] public PlaceHolder(); [C++] public: PlaceHolder(); [VB] Public Sub New() [JScript] public function PlaceHolder(); ChildControlsCreated ClientID Context Controls **EnableViewState Events** HasChildViewState ID **IsTrackingViewState** NamingContainer

1	Parent
2	Site
3	TemplateSourceDirectory
4	UniqueID
5	ViewState
6	ViewStateIgnoresCase
7	Visible
8	PlaceHolderControlBuilder class (System.Web.UI.WebControls)
9	TrackViewState
10	System.Web.UI.WebControls.PlaceHolder
11	PlaceHolderControlBuilder
12	Example Syntax:
13	TrackViewState
14	ControlType
15	FChildrenAsProperties
16	FIsNonParserAccessor
17	HasAspCode
18	ID
19	InDesigner
20	NamingContainerType
21	Parser
22	TagName
23	AllowWhitespaceLiterals
24	

1	[C++] public: bool AllowWhitespaceLiterals();
2	[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
3	[JScript] public override function AllowWhitespaceLiterals(): Boolean; Specifies
4	whether white space literals are allowed.
5	RadioButton class (System.Web.UI.WebControls)
6	ToString
7	
8	
9	Description
10	Represents a radio button control.
11	The System.Web.UI.WebControls.RadioButton server control permits
12	you to intersperse the radio buttons in a group with other content in the page. The
13	buttons are grouped logically because they all share the same
14	System.Web.UI.WebControls.RadioButton.GroupName .
15	RadioButton
16	Example Syntax:
17	ToString
18	
19	[C#] public RadioButton();
20	[C++] public: RadioButton();
21	[VB] Public Sub New()
22	[JScript] public function RadioButton();
23	
24	Description
25	

3

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

т •,• ·	1.	a new instance	C /1
INITIO	1700	O MOTTI INCTOMOO	0 t th/

## System.Web.UI.WebControls.RadioButton class.

AccessKey

Attributes

AutoPostBack

BackColor

BorderColor

BorderStyle

BorderWidth

Checked

ChildControlsCreated

ClientID

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

Enabled

EnableViewState

**Events** 

Font

ForeColor

GroupName

**ToString** 

lee@hayes pilc 509+324+9256

## Description Gets

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the name of the group that the radio button belongs to.

Use this property to specify a grouping of radio buttons to create a mutually exclusive set of controls. The property can be used when only one selection is possible from a list of available options, as in the case of gender.

HasChildViewState

Height

ID

IsTrackingViewState

NamingContainer

Page

**Parent** 

Site

Style

TabIndex

TagKey

TagName

**TemplateSourceDirectory** 

Text

**TextAlign** 

ToolTip

UniqueID

ViewState

1	ViewStateIgnoresCase
2	Visible
3	Width
4	OnPreRender
5	
6	[C#] protected override void OnPreRender(EventArgs e);
7	[C++] protected: void OnPreRender(EventArgs* e);
8	[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)
9	[JScript] protected override function OnPreRender(e : EventArgs);
10	
11	Description
12	This method is invoked just prior to rendering. Register client script for
13	handling postback if onChangeHandler is set.
14	IPostBackDataHandler.LoadPostData
15	
16	[C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
17	NameValueCollection postCollection);
18	[C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
19	NameValueCollection* postCollection);
20	[VB] Function LoadPostData(ByVal postDataKey As String, ByVal
21	postCollection As NameValueCollection) As Boolean Implements
22	IPostBackDataHandler.LoadPostData
23	[JScript] function IPostBackDataHandler.LoadPostData(postDataKey: String,
24	postCollection : NameValueCollection) : Boolean;
25	IPostBackDataHandler.RaisePostDataChangedEvent

[C#] void IPostBackDataHandler.RaisePostDataChangedEvent(); 2 [C++] void IPostBackDataHandler::RaisePostDataChangedEvent(); 3 [VB] Sub RaisePostDataChangedEvent() Implements IPostBackDataHandler.RaisePostDataChangedEvent 5 [JScript] function IPostBackDataHandler.RaisePostDataChangedEvent(); 6 RadioButtonList class (System.Web.UI.WebControls) **TrackViewState** 9 10 Description 11 Represents a list control that encapsulates a group of radio button controls. 12 The System. Web. UI. Web Controls. Radio Button List control provides 13 page developers with a single-selection radio button group that can be dynamically 14 generated via data binding. It contains an 15 System.Web.UI.WebControls.ListControl.Items collection with members that 16 correspond to individual items on the list. To determine which item is selected, 17 test the System. Web. UI. Web Controls. List Control. Selected I tem property of the 18 list. 19 RadioButtonList 20 Example Syntax: 21 **TrackViewState** 22 23 [C#] public RadioButtonList(); 24 [C++] public: RadioButtonList();

[VB] Public Sub New() [JScript] public function RadioButtonList(); 3 Description Initializes a new instance of a System.Web.UI.WebControls.RadioButtonList class. Use this constructor to create and initialize a new instance of the 7 System.Web.UI.WebControls.RadioButtonList class. 8 AccessKey 9 Attributes 10 AutoPostBack 11 BackColor 12 BorderColor 13 BorderStyle 14 **BorderWidth** 15 CellPadding 16 **TrackViewState** 17 18 19 Description 20 Gets or sets the distance (in pixels) between the border and the contents of 21 the table cell. 22 This property only applies when the 23 System.Web.UI.WebControls.RadioButtonList.RepeatLayout property is set 24

to RepeatLayout.Table.

25

1	CellSpacing
2	TrackViewState
3	
4	[C#] public virtual int CellSpacing {get; set;}
5	[C++] public:property virtual int get_CellSpacing();public:property virtua
6	void set_CellSpacing(int);
7	[VB] Overridable Public Property CellSpacing As Integer
8	[JScript] public function get CellSpacing(): int;public function set
9	CellSpacing(int);
10	
11	Description
12	Gets or sets the distance (in pixels) between adjacent table cells.
13	This property only applies when the
14	System.Web.UI.WebControls.RadioButtonList.RepeatLayout property is set
15	to RepeatLayout.Table.
16	ChildControlsCreated
17	ClientID
18	Context
19	Controls
20	ControlStyle
21	ControlStyleCreated
22	CssClass
23	DataMember
24	DataSource
25	DataTextField

18

19

20

21

22

23

24

25

DataTextFormatString DataValueField 2 Enabled 3 EnableViewState **Events** Font ForeColor HasChildViewState Height ID 10 IsTrackingViewState 11 Items 12 NamingContainer 13 Page 14 Parent 15 RepeatColumns 16

TrackViewState

Description

Gets or sets the number of columns to display in the

 ${\bf System. Web. UI. Web Controls. Radio Button List \ control.}$ 

Use this property to specify the number of columns that display items in the **System.Web.UI.WebControls.RadioButtonList** control. If this property is not

set, the System.Web.UI.WebControls.RadioButtonList control will display all items in the list in a single column. 2 RepeatDirection 3 **TrackViewState** 4 5 [C#] public virtual RepeatDirection RepeatDirection {get; set;} 6 [C++] public: property virtual RepeatDirection get RepeatDirection();public: 7 property virtual void set RepeatDirection(RepeatDirection); 8 [VB] Overridable Public Property RepeatDirection As RepeatDirection 9 [JScript] public function get RepeatDirection(): RepeatDirection; public function 10 set RepeatDirection(RepeatDirection); 11 12 Description 13 Gets or sets the direction that the radio buttons within the group are 14 displayed. 15 Use this property to specify the display direction of the 16 System.Web.UI.WebControls.RadioButtonList control. 17 RepeatLayout 18 TrackViewState 19 20 [C#] public virtual RepeatLayout RepeatLayout {get; set;} 21 [C++] public: property virtual RepeatLayout get RepeatLayout();public: 22 property virtual void set RepeatLayout(RepeatLayout); 23 [VB] Overridable Public Property RepeatLayout As RepeatLayout 24 [JScript] public function get RepeatLayout(): RepeatLayout; public function set

25

RepeatLayout(RepeatLayout); 2 Description 3 Gets or sets the layout of radio buttons within the group. 4 Use this property to specify whether the items in the 5 System.Web.UI.WebControls.RadioButtonList control are displayed in a table. 6 If this property is set to **RepeatLayout.Table**, the items in the list are displayed 7 in a table. If this property is set to **RepeatLayout.Flow**, the items in the list are 8 displayed without a table structure. 9 SelectedIndex 10 SelectedItem 11 Site 12 Style 13 TabIndex 14 TagKey 15 **TagName** 16 **TemplateSourceDirectory** 17 TextAlign 18 **TrackViewState** 19 20 21 Description 22 23

Gets or sets the text alignment for the radio buttons within the group.

Use this property to specify whether the text associated with the radio buttons appears on the left or right. If this property is set to TextAlign.Right, the

1	text is displayed to the right of the radio button. If this property is set to
2	TextAlign.Left, the text is displayed to the left of the radio button.
3	ToolTip
4	UniqueID
5	ViewState
6	ViewStateIgnoresCase
7	Visible
8	Width
9	CreateControlStyle
10	
11	[C#] protected override Style CreateControlStyle();
12	[C++] protected: Style* CreateControlStyle();
13	[VB] Overrides Protected Function CreateControlStyle() As Style
14	[JScript] protected override function CreateControlStyle(): Style;
15	
16	Description
17	Render
18	
19	[C#] protected override void Render(HtmlTextWriter writer);
20	[C++] protected: void Render(HtmlTextWriter* writer);
21	[VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)
22	[JScript] protected override function Render(writer: HtmlTextWriter);
23	
24	Description
25	IPostBackDataHandler.LoadPostData

1	
2	[C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
3	NameValueCollection postCollection);
4	[C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
5	NameValueCollection* postCollection);
6	[VB] Function LoadPostData(ByVal postDataKey As String, ByVal
7	postCollection As NameValueCollection) As Boolean Implements
8	IPostBackDataHandler.LoadPostData
9	[JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String
10	postCollection : NameValueCollection) : Boolean;
11	IPostBackDataHandler.RaisePostDataChangedEvent
12	
13	[C#] void IPostBackDataHandler.RaisePostDataChangedEvent();
14	[C++] void IPostBackDataHandler::RaisePostDataChangedEvent();
15	[VB] Sub RaisePostDataChangedEvent() Implements
16	IPostBackDataHandler.RaisePostDataChangedEvent
17	[JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();
18	IRepeatInfoUser.GetItemStyle
19	
20	[C#] Style IRepeatInfoUser.GetItemStyle(ListItemType itemType, int
21	repeatIndex);
22	[C++] Style* IRepeatInfoUser::GetItemStyle(ListItemType itemType, int
23	repeatIndex);
24	[VB] Function GetItemStyle(ByVal itemType As ListItemType, ByVal
25	repeatIndex As Integer) As Style Implements IR eneatInfoLiser GetItemStyle

	[JScript] function IRepeatInfoUser.GetItemStyle(itemType : ListItemType,
	repeatIndex : int) : Style;
	IRepeatInfoUser.RenderItem
	[C#] void IRepeatInfoUser.RenderItem(ListItemType itemType, int repeatIndex,
	RepeatInfo repeatInfo, HtmlTextWriter writer);
	[C++] void IRepeatInfoUser::RenderItem(ListItemType itemType, int
	repeatIndex, RepeatInfo* repeatInfo, HtmlTextWriter* writer);
	[VB] Sub RenderItem(ByVal itemType As ListItemType, ByVal repeatIndex As
	Integer, ByVal repeatInfo As RepeatInfo, ByVal writer As HtmlTextWriter)
	Implements IRepeatInfoUser.RenderItem
	[JScript] function IRepeatInfoUser.RenderItem(itemType: ListItemType,
	repeatIndex: int, repeatInfo: RepeatInfo, writer: HtmlTextWriter);
	RangeValidator class (System.Web.UI.WebControls)
	TrackViewState
	Description
	Checks whether the value of an input control is within a specified range of
	values.
١	

The System.Web.UI.WebControls.RangeValidator control tests whether the value of an input control is within a specified range.

RangeValidator

Example Syntax:

TrackViewState

	1	1
	1	
	2	[C#] public RangeValidator();
	3	[C++] public: RangeValidator();
	4	[VB] Public Sub New()
	5	[JScript] public function RangeValidator();
	6	AccessKey
	7	Attributes
	8	BackColor
	9	BorderColor
t thus that	10	BorderStyle
tuor stave senti stron itanii ilanii ilanii ilanii	11	BorderWidth
The same age	12	ChildControlsCreated
7. 1000	13	ClientID
, and	14	Context
n" door door all door fine	15	Controls
	16	ControlStyle
	17	ControlStyleCreated
	18	ControlToValidate
	19	CssClass
	20	Display
	21	EnableClientScript
	22	Enabled
	23	EnableViewState
	24	ErrorMessage
	25	Events

Font
ForeColor
HasChildViewState
Height
ID
IsTrackingViewState
IsValid
MaximumValue
TrackViewState

## Description

Gets or sets the maximum value of the validation range.

Use the **System.Web.UI.WebControls.RangeValidator.MaximumValue** property to specify the maximum value of the validation range. If the value specified by this property fails to convert to the data type specified by the **System.Web.UI.WebControls.BaseCompareValidator.Type** property, an exception is thrown.

MinimumValue

TrackViewState

[C#] public string MinimumValue {get; set;}

[C++] public: \_\_property String\* get\_MinimumValue();public: \_\_property void

set\_MinimumValue(String\*);

[VB] Public Property MinimumValue As String

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

**ToolTip** 

[JScript] public function get MinimumValue(): String; public function set MinimumValue(String); Description Gets or sets the minimum value of the validation range. Use the System. Web. UI. Web Controls. Range Validator. Minimum Value property to specify the minimum value of the validation range. If the value specified by this property fails to convert to the data type specified by the System.Web.UI.WebControls.BaseCompareValidator.Type property, a trace debug message is generated, the System. Web. UI. WebControls. Base Validator. Is Valid property is set to true, and no additional processing is performed. NamingContainer Page Parent **PropertiesValid** RenderUplevel Site Style TabIndex TagKey **TagName TemplateSourceDirectory Text** 

Type UniqueID ViewState 3 ViewStateIgnoresCase Visible Width AddAttributesToRender 8 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer); 9 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer); 10 [VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As 11 HtmlTextWriter) 12 [JScript] protected override function AddAttributesToRender(writer: 13 HtmlTextWriter); 14 15 Description 16 AddAttributesToRender method AddAttributesToRender method 17 ControlPropertiesValid 18 19 [C#] protected override bool ControlPropertiesValid(); 20 [C++] protected: bool ControlPropertiesValid(); 21 [VB] Overrides Protected Function ControlPropertiesValid() As Boolean 22 [JScript] protected override function ControlPropertiesValid(): Boolean; 23 24 Description

This is a check of properties to determine any errors made by the developer

This is a check of properties to determine any errors made by the developer

EvaluateIsValid

4

2

3

5

6

7 8

9

10

11

13 14

15

16 17

18 19

20

22

21

23 24

25

[C#] protected override bool EvaluateIsValid();

[C++] protected: bool EvaluateIsValid();

[VB] Overrides Protected Function EvaluateIsValid() As Boolean

[JScript] protected override function EvaluateIsValid(): Boolean;

Description

EvaluateIsValid method EvaluateIsValid method

RegularExpressionValidator class (System.Web.UI.WebControls)

Validate

Description

Validates whether the value of an associated input control matches the pattern specified by a regular expression.

The **System.Web.UI.WebControls.RegularExpressionValidator** control checks whether the value of an input control matches a pattern defined by a regular expression. This type of validation allows you to check for predictable sequences of characters, such as those in social security numbers, e-mail addresses, telephone numbers, and postal codes.

RegularExpressionValidator

Example Syntax:

1	Validate
2	
3	[C#] public RegularExpressionValidator();
4	[C++] public: RegularExpressionValidator();
5	[VB] Public Sub New()
6	[JScript] public function RegularExpressionValidator();
7	AccessKey
8	Attributes
9	BackColor
10	BorderColor
11	BorderStyle
12	BorderWidth
13	ChildControlsCreated
14	ClientID
15	Context
16	Controls
17	ControlStyle
18	ControlStyleCreated
19	ControlToValidate
20	CssClass
21	Display
22	EnableClientScript
23	Enabled
24	EnableViewState
25	ErrorMessage

1	Events
2	Font
3	ForeColor
4	HasChildViewState
5	Height
6	ID
7	IsTrackingViewState
8	IsValid
9	NamingContainer
10	Page
11	Parent
12	PropertiesValid
13	RenderUplevel
14	Site
15	Style
16	TabIndex
17	TagKey
18	TagName
19	TemplateSourceDirectory
20	Text
21	ToolTip
22	UniqueID
23	ValidationExpression
24	Validate

Description

2

Gets or sets the regular expression that determines the pattern used to validate a field.

Use this property to specify the pattern used to check for predictable sequences of characters, such as those in social security numbers, e-mail addresses, telephone numbers, and postal codes.

ViewState

ViewStateIgnoresCase

Visible

Width

AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

[C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As

HtmlTextWriter)

[JScript] protected override function AddAttributesToRender(writer:

HtmlTextWriter);

Description

21

22

23

24

25

AddAttributesToRender method AddAttributesToRender method

EvaluateIsValid

1	
2	[C#] protected override bool EvaluateIsValid();
3	[C++] protected: bool EvaluateIsValid();
4	[VB] Overrides Protected Function EvaluateIsValid() As Boolean
5	[JScript] protected override function EvaluateIsValid(): Boolean;
6	
7	Description
8	EvaluateIsValid method EvaluateIsValid method
9	RepeatDirection enumeration (System.Web.UI.WebControls)
10	Validate
11	
12	
13	Description
14	Specifies the direction in which items of a list control are displayed.
15	The System.Web.UI.WebControls.RepeatDirection enumeration
16	represents the different directions in which the items of a list control can be
17	displayed.
18	Validate
19	
20	[C#] public const RepeatDirection Horizontal;
21	[C++] public: const RepeatDirection Horizontal;
22	[VB] Public Const Horizontal As RepeatDirection
23	[JScript] public var Horizontal : RepeatDirection;
24	
25	Description

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Items of a list are displayed horizontally in rows from left to right, then top to bottom, until all items are rendered. Validate [C#] public const RepeatDirection Vertical; [C++] public: const RepeatDirection Vertical; [VB] Public Const Vertical As RepeatDirection [JScript] public var Vertical : RepeatDirection; Description

Items of a list are displayed vertically in columns from top to bottom, and then left to right, until all items are rendered.

Repeater class (System.Web.UI.WebControls)

**ToString** 

Description

A databound list control that allows custom layout by repeating a specified template for each item displayed in the list.

The System. Web. UI. Web Controls. Repeater is a basic templated databound list. It has no built-in layout or styles, so you must explicitly declare all HTML layout, formatting, and style tags within the control's templates.

Repeater

Example Syntax:

**ToString** 

1	
2	[C#] public Repeater();
3	[C++] public: Repeater();
4	[VB] Public Sub New()
5	[JScript] public function Repeater();
6	
7	Description
8	Initializes a new instance of the System.Web.UI.WebControls.Repeater
9	class.
10	AlternatingItemTemplate
11	ToString
12	
13	[C#] public virtual ITemplate AlternatingItemTemplate {get; set;}
14	[C++] public:property virtual ITemplate*
15	get_AlternatingItemTemplate();public:property virtual void
16	set_AlternatingItemTemplate(ITemplate*);
17	[VB] Overridable Public Property AlternatingItemTemplate As ITemplate
18	[JScript] public function get AlternatingItemTemplate(): ITemplate; public
19	function set AlternatingItemTemplate(ITemplate);
20	
21	Description
22	Gets or sets the <b>System.Web.UI.ITemplate</b> that defines how alternating
23	(zero-based odd-indexed) items are rendered.
24	
25	

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use this property to provide a different appearance from the System.Web.UI.WebControls.Repeater.ItemTemplate for alternating items in the System.Web.UI.WebControls.Repeater. ChildControlsCreated ClientID Context Controls **ToString** DataMember **ToString** Description Gets or sets the specific table in the System.Web.UI.WebControls.Repeater.DataSource to bind to the control. If the System. Web. UI. Web Controls. Repeater. Data Source contains multiple tables, use this property to specify the table to bind to the System.Web.UI.WebControls.Repeater . **DataSource** 

**ToString** 

[C#] public virtual object DataSource {get; set;}

[C++] public: property virtual Object\* get DataSource(); public: property

virtual void set\_DataSource(Object\*);

[VB] Overridable Public Property DataSource As Object

1	[JScript] public function get DataSource() : Object;public function set
2	DataSource(Object);
3	
4	Description
5	Gets or sets the data source that provides data for populating the list.
6	Use this property to specify the source of data to populate the
7	System.Web.UI.WebControls.Repeater control. The
8	System.Web.UI.WebControls.Repeater.DataSource can be any
9	System.Collections.IEnumerable derived object such as a
10	System.Data.DataView for accessing databases, an
11	System.Collections.ArrayList, a System.Collections.Hashtable, or an array.
12	EnableViewState
13	Events
14	FooterTemplate
15	ToString
16	
17	
18	Description
19	Gets or sets the <b>System.Web.UI.ITemplate</b> that defines how the control
20	footer is rendered.
21	Use this property to create a template that controls how the footer of a
22	System.Web.UI.WebControls.Repeater control is displayed.
23	HasChildViewState
24	HeaderTemplate
25	ToString

Description

2

3

5

6

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Gets or sets the **System.Web.UI.ITemplate** that defines how the control header is rendered.

Use this property to create a template that controls how the header of a **System.Web.UI.WebControls.Repeater** control is displayed.

ID

IsTrackingViewState

Items

**ToString** 

Description

Gets a collection of **System.Web.UI.WebControls.RepeaterItem** objects in the **System.Web.UI.WebControls.Repeater** .

The System.Web.UI.WebControls.Repeater populates the System.Web.UI.WebControls.Repeater.Items collection by enumerating every object in its System.Web.UI.WebControls.Repeater.DataSource. The System.Web.UI.WebControls.Repeater.Items collection is then used by the System.Web.UI.WebControls.Repeater to render each item in the control.

**ItemTemplate** 

**ToString** 

[C#] public virtual ITemplate ItemTemplate {get; set;}

[C++] public:property virtual ITemplate* get_ItemTemplate();public:
property virtual void set_ItemTemplate(ITemplate*);
[VB] Overridable Public Property ItemTemplate As ITemplate
[JScript] public function get ItemTemplate(): ITemplate;public function set
ItemTemplate(ITemplate);
Description
Gets or sets the System.Web.UI.ITemplate that defines how items are
rendered.
Use this property to create a template that controls how individual items in
the System.Web.UI.WebControls.Repeater are displayed.
NamingContainer
Page
Parent
SeparatorTemplate
ToString
Description
Gets or sets the System.Web.UI.ITemplate that defines how separators
between items are rendered.
Use this property to create a template that controls how separators between
items are displayed.

Site

TemplateSourceDirectory

UniqueID 1 ViewState 2 ViewStateIgnoresCase 3 Visible 4 **ToString** 5 6 7 Description 8 Occurs when a button is clicked in the 9 System.Web.UI.WebControls.Repeater control. 10 This event is raised when a button in the 11 System.Web.UI.WebControls.Repeater is clicked. 12 **ToString** 13 14 [C#] public event RepeaterItemEventHandler ItemCreated; 15 [C++] public: event RepeaterItemEventHandler\* ItemCreated; 16 [VB] Public Event ItemCreated As RepeaterItemEventHandler 17 18 Description 19 Occurs when an item is created in the 20 System.Web.UI.WebControls.Repeater control. 21 This event is raised when an item is created in the 22 System.Web.UI.WebControls.Repeater . 23 **ToString** 24

1	
2	[C#] public event RepeaterItemEventHandler ItemDataBound;
3	[C++] public:event RepeaterItemEventHandler* ItemDataBound;
4	[VB] Public Event ItemDataBound As RepeaterItemEventHandler
5	
6	Description
7	Occurs after an item in the System.Web.UI.WebControls.Repeater is
8	databound but before it is rendered on the page.
9	This event is raised when an item in the
10	System.Web.UI.WebControls.Repeater is databound.
11	CreateChildControls
12	
13	[C#] protected override void CreateChildControls();
14	[C++] protected: void CreateChildControls();
15	[VB] Overrides Protected Sub CreateChildControls()
16	[JScript] protected override function CreateChildControls();
17	
18	Description
19	CreateControlHierarchy
20	
21	[C#] protected virtual void CreateControlHierarchy(bool useDataSource);
22	[C++] protected: virtual void CreateControlHierarchy(bool useDataSource);
23	[VB] Overridable Protected Sub CreateControlHierarchy(ByVal useDataSource
24	As Boolean)
25	[JScript] protected function CreateControlHierarchy(useDataSource : Boolean);

## Description

A protected method. Creates a control hierarchy, with or without the data source as specified. Indicates whether to use the data source or not.

CreateItem

[C#] protected virtual RepeaterItem CreateItem(int itemIndex, ListItemType itemType);

[C++] protected: virtual RepeaterItem\* CreateItem(int itemIndex, ListItemType itemType);

[VB] Overridable Protected Function CreateItem(ByVal itemIndex As Integer, ByVal itemType As ListItemType) As RepeaterItem

[JScript] protected function CreateItem(itemIndex : int, itemType : ListItemType) : RepeaterItem;

## Description

A protected method. Creates a

System.Web.UI.WebControls.RepeaterItem with the specified item type and location within the System.Web.UI.WebControls.Repeater. The specified location within the System.Web.UI.WebControls.Repeater to place the created item. A System.Web.UI.WebControls.ListItemType that represents the specified type of the System.Web.UI.WebControls.Repeater item to create.

**DataBind** 

[C#] public override void DataBind();

[C++] public: void DataBind(); [VB] Overrides Public Sub DataBind() 2 [JScript] public override function DataBind(); 3 4 Description 5 InitializeItem 6 7 [C#] protected virtual void InitializeItem(RepeaterItem item); 8 [C++] protected: virtual void InitializeItem(RepeaterItem\* item); 9 [VB] Overridable Protected Sub InitializeItem(ByVal item As RepeaterItem) 10 [JScript] protected function InitializeItem(item: RepeaterItem); 11 12 Description 13 A protected method. Populates iteratively the specified 14 System.Web.UI.WebControls.RepeaterItem with a sub-hierarchy of child 15 controls. The control to be initialized from an inline template. 16 OnBubbleEvent 17 18 [C#] protected override bool OnBubbleEvent(object sender, EventArgs e); 19 [C++] protected: bool OnBubbleEvent(Object\* sender, EventArgs\* e); 20 [VB] Overrides Protected Function OnBubbleEvent(ByVal sender As Object, 21 ByVal e As EventArgs) As Boolean 22 [JScript] protected override function OnBubbleEvent(sender : Object, e : 23 EventArgs): Boolean; 24 25

1	
2	Description
3	OnDataBinding
4	
5	[C#] protected override void OnDataBinding(EventArgs e);
6	[C++] protected: void OnDataBinding(EventArgs* e);
7	[VB] Overrides Protected Sub OnDataBinding(ByVal e As EventArgs)
8	[JScript] protected override function OnDataBinding(e : EventArgs);
9	
10	Description
11	A protected method. Raises the <b>DataBinding</b> event.
12	OnItemCommand
13	
14	[C#] protected virtual void OnItemCommand(RepeaterCommandEventArgs e);
15	[C++] protected: virtual void OnItemCommand(RepeaterCommandEventArgs* e);
16	[VB] Overridable Protected Sub OnItemCommand(ByVal e As
17	RepeaterCommandEventArgs)
18	[JScript] protected function OnItemCommand(e: RepeaterCommandEventArgs);
19	
20	Description
21	Raises the System.Web.UI.WebControls.Repeater.ItemCommand
22	event.
23	Raising an event invokes the event handler through a delegate. For more
24	information, see . The
25	

1	System.Web.UI.WebControls.RepeaterCommandEventArgs that contains the
2	event data.
3	OnItemCreated
4	
5	[C#] protected virtual void OnItemCreated(RepeaterItemEventArgs e);
6	[C++] protected: virtual void OnItemCreated(RepeaterItemEventArgs* e);
7	[VB] Overridable Protected Sub OnItemCreated(ByVal e As
8	RepeaterItemEventArgs)
9	[JScript] protected function OnItemCreated(e: RepeaterItemEventArgs);
10	
11	Description
12	Raises the System.Web.UI.WebControls.Repeater.ItemCreated event.
13	Raising an event invokes the event handler through a delegate. For more
14	information, see . The System. Web. UI. Web Controls. Repeater I tem Event Args
15	that contains the event data.
16	OnItemDataBound
17	
18	[C#] protected virtual void OnItemDataBound(RepeaterItemEventArgs e);
19	[C++] protected: virtual void OnItemDataBound(RepeaterItemEventArgs* e);
20	[VB] Overridable Protected Sub OnItemDataBound(ByVal e As
21	RepeaterItemEventArgs)
22	[JScript] protected function OnItemDataBound(e: RepeaterItemEventArgs);
23	
24	Description
25	

Raises the **System.Web.UI.WebControls.Repeater.ItemDataBound** event.

Raising an event invokes the event handler through a delegate. For more information, see . The **System.Web.UI.WebControls.RepeaterItemEventArgs** that contains the event data.

RepeaterCommandEventArgs class (System.Web.UI.WebControls)
TrackViewState

## Description

Provides data for the

System.Web.UI.WebControls.Repeater.ItemCommand event of a System.Web.UI.WebControls.Repeater . This class cannot be inherited.

For a list of initial property values for an instance of 
System.Web.UI.WebControls.RepeaterCommandEventArgs, see the 
System.Web.UI.WebControls.RepeaterCommandEventArgs.#ctor 
constructor. For more information about handling events, see .

RepeaterCommandEventArgs

Example Syntax:

**TrackViewState** 

[C#] public RepeaterCommandEventArgs(RepeaterItem item, object commandSource, CommandEventArgs originalArgs);
[C++] public: RepeaterCommandEventArgs(RepeaterItem\* item, Object\* commandSource, CommandEventArgs\* originalArgs);

[VB] Public Sub New(ByVal item As RepeaterItem, ByVal commandSource As
Object, ByVal originalArgs As CommandEventArgs)
[JScript] public function RepeaterCommandEventArgs(item: RepeaterItem,
commandSource : Object, originalArgs : CommandEventArgs);
Description
Initializes a new instance of the
System.Web.UI.WebControls.RepeaterCommandEventArgs class.
The following table shows initial property values for an instance of
$System. Web. UI. Web Controls. Repeater Command Event Args. \ A$
System.Web.UI.WebControls.RepeaterItem that represents an item in the
System.Web.UI.WebControls.Repeater. The
System.Web.UI.WebControls.RepeaterCommandEventArgs.Item property is
set to this value. The command source. The
System. We b. UI. We b Controls. Repeater Command Event Args. Command Source and Sourc
e property is set to this value. The original event arguments.
CommandArgument
CommandName
CommandSource
TrackViewState
Description
Gets the source of the command.
Use this property to determine the source of the command.

Item

TrackViewState

[C#] public RepeaterItem Item {get;}

[C++] public: property RepeaterItem\* get\_Item();

[VB] Public ReadOnly Property Item As RepeaterItem

[JScript] public function get Item(): RepeaterItem;

Description

Gets the **System.Web.UI.WebControls.RepeaterItem** associated with the event.

Use this property to access the properties of the

System.Web.UI.WebControls.RepeaterItem associated with this event.

RepeaterCommandEventHandler delegate (System.Web.UI.WebControls)
ToString

Description

Represents the method that will handle the

System.Web.UI.WebControls.Repeater.ItemCommand event of a

System.Web.UI.WebControls.Repeater. The source of the event. A

System.Web.UI.WebControls.RepeaterCommandEventArgs that contains the

event data.

23

24

When you create a

System.Web.UI.WebControls.RepeaterCommandEventHandler delegate, you

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

identify the method that will handle the event. To associate the event with your event handler, add an instance of the delegate to the event. The event handler is called whenever the event occurs, unless you remove the delegate. For more information about event handler delegates, see .

RepeaterItem class (System.Web.UI.WebControls)

**ToString** 

Description

Represents an item in the **System.Web.UI.WebControls.Repeater** control.

A System.Web.UI.WebControls.RepeaterItem object represents an item in the System.Web.UI.WebControls.Repeater control, such as the heading section, footer section, or a data item.

RepeaterItem

Example Syntax:

**ToString** 

[C#] public RepeaterItem(int itemIndex, ListItemType itemType);

[C++] public: RepeaterItem(int itemIndex, ListItemType itemType);

[VB] Public Sub New(ByVal itemIndex As Integer, ByVal itemType As

ListItemType)

[JScript] public function RepeaterItem(itemIndex: int, itemType: ListItemType);

Description

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the

System.Web.UI.WebControls.RepeaterItem class.

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.RepeaterItem class. The index of the item in the System.Web.UI.WebControls.Repeater control from the

System.Web.UI.WebControls.Repeater.Items collection of the control. One of the System.Web.UI.WebControls.ListItemType values.

ChildControlsCreated

ClientID

Context

Controls

DataItem

**ToString** 

Description

Gets or sets a data item associated with the

System.Web.UI.WebControls.RepeaterItem object in the

System.Web.UI.WebControls.Repeater control.

Use the **System.Web.UI.WebControls.RepeaterItem.DataItem** property to specify or determine the properties of a data item associated with the

System.Web.UI.WebControls.RepeaterItem object in the

System.Web.UI.WebControls.Repeater control.

EnableViewState

**Events** 

HasChildViewState ID Is Tracking View State3 ItemIndex **ToString** 5 7 Description 8 Gets the index of the item in the System.Web.UI.WebControls.Repeater 9 control from the System. Web. UI. Web Controls. Repeater. Items collection of the 10 control. 11 Use the System.Web.UI.WebControls.RepeaterItem.ItemIndex property 12 to determine the index number of the item in the 13 System.Web.UI.WebControls.Repeater control from the 14 System. Web. UI. Web Controls. Repeater. Items collection of the control. 15 ItemType 16 **ToString** 17 18 [C#] public virtual ListItemType ItemType {get;} 19 [C++] public: \_\_property virtual ListItemType get\_ItemType(); 20 [VB] Overridable Public ReadOnly Property ItemType As ListItemType 21 [JScript] public function get ItemType(): ListItemType; 22 23 Description 24 25

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets the type of the item in the System.Web.UI.WebControls.Repeater control. Use the System.Web.UI.WebControls.RepeaterItem.ItemType property to determine the type of an item in the System.Web.UI.WebControls.Repeater control. The following table lists the different item types. NamingContainer Page Parent Site **TemplateSourceDirectory** UniqueID ViewState ViewStateIgnoresCase Visible **OnBubbleEvent** [C#] protected override bool OnBubbleEvent(object source, EventArgs e); [C++] protected: bool OnBubbleEvent(Object\* source, EventArgs\* e); [VB] Overrides Protected Function OnBubbleEvent(ByVal source As Object, ByVal e As EventArgs) As Boolean [JScript] protected override function OnBubbleEvent(source : Object, e : EventArgs): Boolean; Description

RepeaterItemCollection class (System.Web.UI.WebControls)

7T 1 T	7*	$\alpha_{+-+-}$
I TOOK	/ 1033/	NT2TA
Track	LO W	Jun

3

2

5

7

8

9 10

11

12

14 15

17

16

18

20

19

21

22

2324

25

Description

Represents a collection of **System.Web.UI.WebControls.RepeaterItem** objects in the **System.Web.UI.WebControls.Repeater** control. This class cannot be inherited.

The System.Web.UI.WebControls.RepeaterItemCollection class represents a collection of System.Web.UI.WebControls.RepeaterItem objects, which in turn represent the data items in a

System.Web.UI.WebControls.Repeater control. To programmatically retrieve System.Web.UI.WebControls.RepeaterItem objects from a

System.Web.UI.WebControls.Repeater control, use one of the following methods: Use the indexer to get a single

**System.Web.UI.WebControls.RepeaterItem** object from the collection using array notation.

RepeaterItemCollection

Example Syntax:

TrackViewState

[C#] public RepeaterItemCollection(ArrayList items);

[C++] public: RepeaterItemCollection(ArrayList\* items);

[VB] Public Sub New(ByVal items As ArrayList)

[JScript] public function RepeaterItemCollection(items : ArrayList);

Description

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Initializes a new instance of the

 ${\bf System. Web. UI. Web Controls. Repeater Item Collection\ class.}$ 

Use this constructor to create and initialize a new instance of the System.Web.UI.WebControls.RepeaterItemCollection class. A System.Collections.ArrayList object that contains the items with which to initialize the collection.

Count

TrackViewState

[C#] public int Count {get;}

[C++] public: property int get\_Count();

[VB] Public ReadOnly Property Count As Integer

[JScript] public function get Count(): int;

Description

Gets the number of **System.Web.UI.WebControls.RepeaterItem** objects in the collection.

Use the **System.Web.UI.WebControls.RepeaterItemCollection.Count** property to determine the number of

System.Web.UI.WebControls.RepeaterItem objects in the

 ${\bf System. Web. UI. WebControls. Repeater Item Collection}\ collection.\ The$ 

System.Web.UI.WebControls.RepeaterItemCollection.Count property is

lee@hayes pilc 509-324-9256

2062

1	commonly used when iterating through the collection to determine the upper
2	bound of the collection.
3	IsReadOnly
4	TrackViewState
5	
6	[C#] public bool IsReadOnly {get;}
7	[C++] public:property bool get_IsReadOnly();
8	[VB] Public ReadOnly Property IsReadOnly As Boolean
9	[JScript] public function get IsReadOnly(): Boolean;
10	•
11	Description
12	Gets a value that indicates whether the
13	System.Web.UI.WebControls.RepeaterItem objects in the
14	System.Web.UI.WebControls.RepeaterItemCollection can be modified.
15	This property always returns false to indicate that the
16	System.Web.UI.WebControls.RepeaterItemCollection can be written to in all
17	cases.
18	IsSynchronized
19	TrackViewState
20	
21	[C#] public bool IsSynchronized {get;}
22	[C++] public:property bool get_IsSynchronized();
23	[VB] Public ReadOnly Property IsSynchronized As Boolean
24	[JScript] public function get IsSynchronized(): Boolean;
25	

Description

1

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets a value indicating whether access to the System.Web.UI.WebControls.RepeaterItemCollection is synchronized (thread-safe).

This property is derived from the **System.Collections.ICollection** class and is overridden to always return **false**.

Item

**TrackViewState** 

[C#] public RepeaterItem this[int index] {get;}

[C++] public: \_\_property RepeaterItem\* get\_Item(int index);

[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As

RepeaterItem

[JScript] returnValue = RepeaterItemCollectionObject.Item(index);

Description

Gets the **System.Web.UI.WebControls.RepeaterItem** object at the specified index in the collection.

Use this indexer to get a **System.Web.UI.WebControls.RepeaterItem** object from the **System.Web.UI.WebControls.RepeaterItemCollection** at the specified index, using array notation. The zero-based index of the

System. Web. UI. Web Controls. Repeater I tem object to retrieve in the collection.

SyncRoot

**TrackViewState** 

1 [C#] public object SyncRoot {get;} [C++] public: \_\_property Object\* get\_SyncRoot(); 3 [VB] Public ReadOnly Property SyncRoot As Object [JScript] public function get SyncRoot(): Object; 6 Description 7 Gets the object that can be used to synchronize access to the 8  ${\bf System. Web. UI. WebControls. Repeater Item Collection}\ collection.$ 9 The object returned in this implementation is the 10 System.Web.UI.WebControls.RepeaterItemCollection object itself. 11 СоруТо 12 13 [C#] public void CopyTo(Array array, int index); 14 [C++] public: \_\_sealed void CopyTo(Array\* array, int index); 15 [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As 16 Integer) 17 [JScript] public function CopyTo(array : Array, index : int); 18 19 Description 20 Copies all the items from this 21  ${\bf System. Web. UI. Web Controls. Repeater I tem Collection}\ \ {\bf to}\ \ {\bf the}\ \ {\bf specified}$ 22 System.Array object, starting at the specified index in the System.Array object. 23 Use this method to copy the contents of the 24 System.Web.UI.WebControls.RepeaterItemCollection into the specified 25

1	System. Array object, starting at the specified index. A zero-based System. Array
2	object that receives the copied items from the
3	System.Web.UI.WebControls.RepeaterItemCollection. The first position in the
4	specified System.Array object to receive the copied contents.
5	GetEnumerator
6	
7	[C#] public IEnumerator GetEnumerator();
8	[C++] public:sealed IEnumerator* GetEnumerator();
9	[VB] NotOverridable Public Function GetEnumerator() As IEnumerator
10	[JScript] public function GetEnumerator(): IEnumerator;
11	
12	Description
13	Returns a System.Collections.IEnumerator interface that contains all
14	System.Web.UI.WebControls.RepeaterItem objects in the
15	System.Web.UI.WebControls.RepeaterItemCollection .
16	Return Value: A System.Collections.IEnumerator interface that contains all
17	System.Web.UI.WebControls.RepeaterItem objects in the
18	System.Web.UI.WebControls.RepeaterItemCollection .
19	Use this method to create a System.Collections.IEnumerator that can be
20	easily iterated through to get each item in the
21	System.Web.UI.WebControls.RepeaterItemCollection .
22	RepeaterItemEventArgs class (System.Web.UI.WebControls)
23	ToString
24	

System.Web.UI.WebControls.Repeater ItemEventArgs , see the System.Web.UI.WebControls.RepeaterItemEventArgs.#ctor RepeaterItemEventArgs  Example Syntax: ToString  [C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item: RepeaterItem)  Description Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class. The following table shows initial property values for an instance of the	2	
System.Web.UI.WebControls.Repeater.ItemCreated and System.Web.UI.WebControls.Repeater.ItemDataBound even System.Web.UI.WebControls.Repeater.  For a list of initial property values for an instance of System.Web.UI.WebControls.RepeaterItemEventArgs, see th System.Web.UI.WebControls.RepeaterItemEventArgs.#ctor RepeaterItemEventArgs  Example Syntax: ToString  [C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item: Repeate  Description Initializes a new instance of the System.Web.UI.WebControls.RepeaterItemEventArgs class. The following table shows initial property values for an in-	3	Description
System.Web.UI.WebControls.Repeater.ItemDataBound even System.Web.UI.WebControls.Repeater.  For a list of initial property values for an instance of System.Web.UI.WebControls.RepeaterItemEventArgs, see the System.Web.UI.WebControls.RepeaterItemEventArgs.#ctor RepeaterItemEventArgs  Example Syntax: ToString  [C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item: Repeater  Description  Initializes a new instance of the System.Web.UI.WebControls.RepeaterItemEventArgs class. The following table shows initial property values for an in-	4	Provides data for the
System.Web.UI.WebControls.Repeater ItemEventArgs , see the System.Web.UI.WebControls.RepeaterItemEventArgs.#ctor RepeaterItemEventArgs  Example Syntax: ToString  [C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item: RepeaterItem)  Description Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class. The following table shows initial property values for an instance of the	5	System.Web.UI.WebControls.Repeater.ItemCreated and
For a list of initial property values for an instance of  System.Web.UI.WebControls.RepeaterItemEventArgs, see fil  System.Web.UI.WebControls.RepeaterItemEventArgs.#ctor  RepeaterItemEventArgs  Example Syntax:  ToString  [C#] public RepeaterItemEventArgs(RepeaterItem item);  [C++] public: RepeaterItemEventArgs(RepeaterItem* item);  [VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item: Repeater  Description  Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an in-	6	System.Web.UI.WebControls.Repeater.ItemDataBound events of a
System.Web.UI.WebControls.RepeaterItemEventArgs, see the System.Web.UI.WebControls.RepeaterItemEventArgs.#ctor RepeaterItemEventArgs  Example Syntax: ToString  [C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem) [JScript] public function RepeaterItemEventArgs(item: Repeater Description Initializes a new instance of the System.Web.UI.WebControls.RepeaterItemEventArgs class. The following table shows initial property values for an instance of the system.Web.UI.WebControls.RepeaterItemEventArgs class.	7	System.Web.UI.WebControls.Repeater .
System.Web.UI.WebControls.RepeaterItemEventArgs.#ctor  RepeaterItemEventArgs  Example Syntax:  ToString  [C#] public RepeaterItemEventArgs(RepeaterItem item);  [C++] public: RepeaterItemEventArgs(RepeaterItem* item);  [VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item: Repeate)  Description  Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an in-	8	For a list of initial property values for an instance of
RepeaterItemEventArgs  Example Syntax:  ToString  [C#] public RepeaterItemEventArgs(RepeaterItem item);  [C++] public: RepeaterItemEventArgs(RepeaterItem* item);  [VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item: Repeate  Description  Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an in	9	${\bf System. Web. UI. WebControls. Repeater I tem Event Args}\ ,  {\it see the}$
[C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem) [JScript] public function RepeaterItemEventArgs(item: Repeater Property item)  Description Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an initializes and item.	10	$System. Web. UI. Web Controls. Repeater I tem Event Args. \#ctor \ constructor.$
[C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem) [JScript] public function RepeaterItemEventArgs(item: Repeater Property item)  Description Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an initializes and item.	11	RepeaterItemEventArgs
[C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem) [JScript] public function RepeaterItemEventArgs(item: Repeater Property item)  Description Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an initializes and item.	12	Example Syntax:
[C#] public RepeaterItemEventArgs(RepeaterItem item); [C++] public: RepeaterItemEventArgs(RepeaterItem* item); [VB] Public Sub New(ByVal item As RepeaterItem) [JScript] public function RepeaterItemEventArgs(item: Repeater Property item)  Description Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an initializes and item.	13	ToString
[VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item : Repeate  Description  Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an initial property value value value value value value value value value value value value value value value value value value valu		
[VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item : Repeate  Description  Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an initial property value value value value value value value value value value value value value value value value value value valu	15	[C#] public RepeaterItemEventArgs(RepeaterItem item);
[VB] Public Sub New(ByVal item As RepeaterItem)  [JScript] public function RepeaterItemEventArgs(item : Repeate  Description  Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an initial property value value value value value value value value value value value value value value value value value value valu	16	[C++] public: RepeaterItemEventArgs(RepeaterItem* item);
Description Initializes a new instance of the System.Web.UI.WebControls.RepeaterItemEventArgs class. The following table shows initial property values for an in	11	[VB] Public Sub New(ByVal item As RepeaterItem)
Description Initializes a new instance of the System.Web.UI.WebControls.RepeaterItemEventArgs class. The following table shows initial property values for an in	18	[JScript] public function RepeaterItemEventArgs(item: RepeaterItem);
Initializes a new instance of the  System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an in-	19	
System.Web.UI.WebControls.RepeaterItemEventArgs class.  The following table shows initial property values for an in	20	Description
The following table shows initial property values for an in	21	Initializes a new instance of the
	22	System.Web.UI.WebControls.RepeaterItemEventArgs class.
System.Web.UI.WebControls.RepeaterItemEventArgs . The	23	The following table shows initial property values for an instance of
- T	24	System.Web.UI.WebControls.RepeaterItemEventArgs . The
System.Web.UI.WebControls.RepeaterItem associated with t	25	System.Web.UI.WebControls.RepeaterItem associated with the event. The

	1	System.Web.UI.WebControls.RepeaterItemEventArgs.Item property is set to
	2	this value.
	3	Item
	4	ToString
	5	
	6	[C#] public RepeaterItem Item {get;}
	7	[C++] public:property RepeaterItem* get_Item();
	8	[VB] Public ReadOnly Property Item As RepeaterItem
	9	[JScript] public function get Item(): RepeaterItem;
	10	
	11	Description
# # # # # # # # # # # # # # # # # # #	12	Gets the System.Web.UI.WebControls.RepeaterItem associated with the
in the time that the time that the time	13	event.
	14	Use this property to access the properties of the
	15	System.Web.UI.WebControls.RepeaterItem associated with this event.
	16	RepeaterItemEventHandler delegate (System.Web.UI.WebControls)
:: :::2	17	ToString
	18	
	19	
	20	Description
	21	Represents the method that will handle the
	22	
	23	11 -
	24	System.Web.UI.WebControls.Repeater . The source of the event. A
	25	

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

System.Web.UI.WebControls.RepeaterItemEventArgs that contains the event data.

When you create a

System.Web.UI.WebControls.RepeaterItemEventHandler delegate, you identify the method that will handle the event. To associate the event with your event handler, add an instance of the delegate to the event. The event handler is called whenever the event occurs, unless you remove the delegate. For more information about event handler delegates, see .

RepeatInfo class (System.Web.UI.WebControls)
ToString

Description

Defines the information used to render a list of items using a System.Web.UI.WebControls.Repeater .

RepeatInfo

Example Syntax:

**ToString** 

[C#] public RepeatInfo();

[C++] public: RepeatInfo();

[VB] Public Sub New()

[JScript] public function RepeatInfo();

Description

1	Initializes a new instance of the
2	System.Web.UI.WebControls.RepeatInfo class. This class is not inheritable.
3	OuterTableImplied
4	ToString
5	
6	[C#] public bool OuterTableImplied {get; set;}
7	[C++] public:property bool get_OuterTableImplied();public:property void
8	set_OuterTableImplied(bool);
9	[VB] Public Property OuterTableImplied As Boolean
10	[JScript] public function get OuterTableImplied(): Boolean; public function set
11	OuterTableImplied(Boolean);
12	
13	Description
14	Indicates whether an outer table is implied for the items.
15	RepeatColumns
16	ToString
17	
18	[C#] public int RepeatColumns {get; set;}
19	[C++] public:property int get_RepeatColumns();public:property void
20	set_RepeatColumns(int);
21	[VB] Public Property RepeatColumns As Integer
22	[JScript] public function get RepeatColumns(): int;public function set
23	RepeatColumns(int);
24	
25	Description

Indicates the column count of items. 1 RepeatDirection 2 **ToString** 3 [C#] public RepeatDirection RepeatDirection {get; set;} [C++] public: \_\_property RepeatDirection get\_RepeatDirection();public: property void set\_RepeatDirection(RepeatDirection); 7 [VB] Public Property RepeatDirection As RepeatDirection [JScript] public function get RepeatDirection(): RepeatDirection; public function set RepeatDirection(RepeatDirection); 10 11 Description 12 Indicates the direction of flow of items. 13 RepeatLayout 14 **ToString** 15 16 [C#] public RepeatLayout RepeatLayout {get; set;} 17 [C++] public: \_\_property RepeatLayout get\_RepeatLayout();public: \_\_property 18 void set RepeatLayout(RepeatLayout); 19 [VB] Public Property RepeatLayout As RepeatLayout 20 [JScript] public function get RepeatLayout() : RepeatLayout;public function set 21 RepeatLayout(RepeatLayout); 22 23 Description 24 Indicates the layout of items. 25

Rend	lerRe	peater
KOHO	CITC	poutor

3

4

7

8

9

11

13

12

15

14

17

18

16

19

20

21

23

24

25

[C#] public void RenderRepeater(HtmlTextWriter writer, IRepeatInfoUser user, Style controlStyle, WebControl baseControl);

[C++] public: void RenderRepeater(HtmlTextWriter\* writer, IRepeatInfoUser\* user, Style\* controlStyle, WebControl\* baseControl);

[VB] Public Sub RenderRepeater(ByVal writer As HtmlTextWriter, ByVal user As IRepeatInfoUser, ByVal controlStyle As Style, ByVal baseControl As WebControl)

[JScript] public function RenderRepeater(writer: HtmlTextWriter, user:

IRepeatInfoUser, control Style: Style, baseControl: WebControl);

# Description

Renders the Repeater with the specified information. The output stream that renders HTML content to the client. The control style to copy. The control to copy base attributes from.

RepeatLayout enumeration (System.Web.UI.WebControls)
ToString

# Description

Specifies the layout of items in a list control.

The **System.Web.UI.WebControls.RepeatLayout** enumeration represents the different layout options for a list control.

**ToString** 

[C#] public const RepeatLayout Flow; [C++] public: const RepeatLayout Flow; [VB] Public Const Flow As RepeatLayout [JScript] public var Flow: RepeatLayout; 6 Description Items are displayed without a table structure. 8 **ToString** 9 10 [C#] public const RepeatLayout Table; 11 [C++] public: const RepeatLayout Table; 12 [VB] Public Const Table As RepeatLayout 13 [JScript] public var Table : RepeatLayout; 14 15 Description 16 Items are displayed in a table. 17 RequiredFieldValidator class (System.Web.UI.WebControls) 18 **ToString** 19 20 21 Description 22 Makes the associated input control a required field. 23 Use this control to make an input control a required field. The input control 24 fails validation if its value does not change from the 25

Display

	1	EnableClientScript
	2	Enabled
	3	EnableViewState
	4	ErrorMessage
	5	Events
	6	Font
	7	ForeColor
	8	HasChildViewState
	9	Height
	10	ID
	11	InitialValue
	12	ToString
n in the second	13	
	14	
	15	Description
: 1000g : 1000g : 1000g : 1000g	16	Gets or sets the initial value of the associated input control.
us Transform Transform	17	Use this property to specify the initial value of the input control.
	18	IsTrackingViewState
	19	IsValid
	20	NamingContainer
	21	Page

Parent

Site

PropertiesValid

RenderUplevel

22

23

24

1	Style
2	TabIndex
3	TagKey
4	TagName
5	TemplateSourceDirectory
6	Text
7	ToolTip
8	UniqueID
9	ViewState
10	ViewStateIgnoresCase
11	Visible
12	Width
13	AddAttributesToRender
14	
15	[C#] protected override void AddAttributesToRender(HtmlTextWriter writer)
16	[C++] protected: void AddAttributesToRender(HtmlTextWriter* writer);
17	[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As
18	HtmlTextWriter)
19	[JScript] protected override function AddAttributesToRender(writer:
20	HtmlTextWriter);
21	
22	Description
23	AddAttributesToRender method AddAttributesToRender method
24	EvaluateIsValid
25	

1	
2	[C#] protected override bool EvaluateIsValid();
3	[C++] protected: bool EvaluateIsValid();
4	[VB] Overrides Protected Function EvaluateIsValid() As Boolean
5	[JScript] protected override function EvaluateIsValid(): Boolean;
6	
7	Description
8	EvaluateIsValid method EvaluateIsValid method
9	Table.RowControlCollection class (System.Web.UI.WebControls)
10	Validate
11	
12	
13	Description
14	
15	Count
16	IsReadOnly
17	IsSynchronized
18	Item
19	Owner
20	SyncRoot
21	Add
22	
23	[C#] public override void Add(Control child);
24	[C++] public: void Add(Control* child);
25	[VB] Overrides Public Sub Add(ByVal child As Control)

Ī
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
26

[JScript] public override function Add(child : Control);

Description

2

Adds the specified **System.Web.UI.Control** object to the collection. The new control is added to the end of the array.

AddAt

[C#] public override void AddAt(int index, Control child);

[C++] public: void AddAt(int index, Control\* child);

[VB] Overrides Public Sub AddAt(ByVal index As Integer, ByVal child As

Control)

[JScript] public override function AddAt(index : int, child : Control);

# Description

Adds the specified **System.Web.UI.Control** object to the collection. The new control is added to the array at the specified index location. The location in the array to add the child control. The **Control** object to add to the collection.

SelectedDatesCollection class (System.Web.UI.WebControls)

ToString

### Description

Encapsulates a collection of **System.DateTime** objects that represent the selected dates in a **System.Web.UI.WebControls.Calendar** control. This class cannot be inherited.

	1	Use this class to programmatically manage a collection of
	2	System.DateTime objects that represent the selected dates in a
	3	System.Web.UI.WebControls.Calendar control. This class is commonly used to
	4	add or remove dates.
	5	SelectedDatesCollection
	6	Example Syntax:
	7	ToString
	8	
	9	[C#] public SelectedDatesCollection(ArrayList dateList);
A COLUMN TO THE PARTY OF THE PA	10	[C++] public: SelectedDatesCollection(ArrayList* dateList);
	11	[VB] Public Sub New(ByVal dateList As ArrayList)
ant that	12	[JScript] public function SelectedDatesCollection(dateList : ArrayList);
i i	13	
Third in the	14	Description
	15	Initializes a new instance of the
	16	System.Web.UI.WebControls.SelectedDatesCollection class with the specified
	17	date list.
	18	Use this constructor to create a new instance of the
	19	System.Web.UI.WebControls.SelectedDatesCollection class. A
	20	System.Collections.ArrayList that represents a collection of dates.
	21	Count
	22	ToString
	23	
	24	[C#] public int Count {get;}
	25	[C++] public:property int get_Count();

1	[VB] Public ReadOnly Property Count As Integer
2	[JScript] public function get Count(): int;
3	
4	Description
5	Gets the number of <b>System.DateTime</b> objects in the
6	System.Web.UI.WebControls.SelectedDatesCollection .
7	Use this property to determine the number of dates in the
8	System.Web.UI.WebControls.SelectedDatesCollection . The
9	System.Web.UI.WebControls.SelectedDatesCollection.Count property is often
10	used when iterating through the collection to determine the upper bound.
11	IsReadOnly
12	ToString
13	
14	[C#] public bool IsReadOnly {get;}
15	[C++] public:property bool get_IsReadOnly();
16	[VB] Public ReadOnly Property IsReadOnly As Boolean
17	[JScript] public function get IsReadOnly(): Boolean;
18	
19	Description
20	Gets a value indicating whether the
21	System.Web.UI.WebControls.SelectedDatesCollection is read-only.
22	This property always returns false to indicate that the
23	System.Web.UI.WebControls.SelectedDatesCollection can be written to in all
24	cases.
25	IsSynchronized

1	ToString
2	
3	[C#] public bool IsSynchronized {get;}
4	[C++] public:property bool get_IsSynchronized();
5	[VB] Public ReadOnly Property IsSynchronized As Boolean
6	[JScript] public function get IsSynchronized(): Boolean;
7	
8	Description
9	Gets a value indicating whether access to the
10	System.Web.UI.WebControls.SelectedDatesCollection is synchronized (thread-
11	safe).
12	This property is derived from System.Collections.ICollection and is
13	overridden to always return false.
14	Item
15	ToString
16	
17	[C#] public DateTime this[int index] {get;}
18	[C++] public:property DateTime get_Item(int index);
19	[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As
20	DateTime
21	[JScript] returnValue = SelectedDatesCollectionObject.Item(index);
22	
23	Description
24	Gets a System.DateTime at the specified index in the
25	System.Web.UI.WebControls.SelectedDatesCollection .

Description

Use this indexer to get an individual System.DateTime in the
System.Web.UI.WebControls.SelectedDatesCollection at the specified index
using simple array notation. An ordinal index value that specifies which
System.DateTime to return.
SyncRoot
ToString
[C#] public object SyncRoot {get;}
[C++] public:property Object* get_SyncRoot();
[VB] Public ReadOnly Property SyncRoot As Object
[JScript] public function get SyncRoot(): Object;
Description
Gets the object that can be used to synchronize access to the
$System. Web. UI. Web Controls. Selected Dates Collection \ .$
The object returned in this implementation is the
System.Web.UI.WebControls.SelectedDatesCollection object itself.
Add
[C#] public void Add(DateTime date);
[C++] public: void Add(DateTime date);
[VB] Public Sub Add(ByVal date As DateTime)
[JScript] public function Add(date : DateTime);

1	Appends the specified System.DateTime to the end of the
2	System.Web.UI.WebControls.SelectedDatesCollection .
3	Use this method to add the specified System.DateTime to the end of the
4	$System. We b. UI. We b Controls. Selected Dates Collection\ .\ The\ System. Date Time$
5	to add to the collection.
6	Clear
7	
8	[C#] public void Clear();
9	[C++] public: void Clear();
10	[VB] Public Sub Clear()
11	[JScript] public function Clear();
12	
13	Description
14	Removes all System.DateTime controls from the collection.
15	Use this method to remove all System.DateTime objects from the
16	System.Web.UI.WebControls.SelectedDatesCollection and set the
17	System.Web.UI.WebControls.SelectedDatesCollection.Count property to 0.
18	Contains
19	
20	[C#] public bool Contains(DateTime date);
21	[C++] public: bool Contains(DateTime date);
22	[VB] Public Function Contains(ByVal date As DateTime) As Boolean
23	[JScript] public function Contains(date : DateTime) : Boolean;
24	-
25	Description

1	Returns a value indicating whether the
2	System.Web.UI.WebControls.SelectedDatesCollection contains the specified
3	System.DateTime object.
4	Return Value: true if the
5	System.Web.UI.WebControls.SelectedDatesCollection contains the specified
6	System.DateTime; otherwise, false.
7	Use this method to determine whether the
8	System.Web.UI.WebControls.SelectedDatesCollection contains the specified
9	System.DateTime . The System.DateTime object to search for in the
10	System.Web.UI.WebControls.SelectedDatesCollection.
11	СоруТо
12	
13	[C#] public void CopyTo(Array array, int index);
14	[C++] public:sealed void CopyTo(Array* array, int index);
15	[VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
16	Integer)
17	[JScript] public function CopyTo(array: Array, index: int);
18	
19	Description
20	Copies the items from the
21	System.Web.UI.WebControls.SelectedDatesCollection to the specified
22	System.Array, starting with the specified index.
23	Use this method to copy the contents of the
24	System.Web.UI.WebControls.SelectedDatesCollection into the specified
25	System.Array, starting at the specified index. A zero-based System.Array that

receives the copied items from the System.Web.UI.WebControls.SelectedDatesCollection. The first index in the 2 specified System.Array to receive the items. 3 GetEnumerator 5 [C#] public IEnumerator GetEnumerator(); 6 [C++] public: sealed IEnumerator\* GetEnumerator(); 7 [VB] NotOverridable Public Function GetEnumerator() As IEnumerator 8 [JScript] public function GetEnumerator(): IEnumerator; 9 10 Description 11 Returns a System. Collections. IE numerator that contains all 12 System.DateTime objects within the 13  $System. Web. UI. Web Controls. Selected Dates Collection \ .$ 14 Return Value: A System.Collections.IEnumerator that contains all 15 System.DateTime objects within the 16 System.Web.UI.WebControls.SelectedDatesCollection . 17 Use this method to create a System. Collections. IE numerator that can be 18 easily iterated through to get each item in the 19 System.Web.UI.WebControls.SelectedDatesCollection . 20 Remove 21 22 [C#] public void Remove(DateTime date); 23 [C++] public: void Remove(DateTime date); 24 [VB] Public Sub Remove(ByVal date As DateTime)

[JScript] public function Remove(date : DateTime); 2 Description 3 Removes the specified System.DateTime object from the System. Web. UI. Web Controls. Selected Dates Collection.5 Use this method to remove the specified System.DateTime from the 6  $System. Web. UI. Web Controls. Selected Dates Collection\ .\ The\ System. Date Time$ 7 object to remove from the 8 System.Web.UI.WebControls.SelectedDatesCollection. 9 SelectRange 10 11 [C#] public void SelectRange(DateTime fromDate, DateTime toDate); 12 [C++] public: void SelectRange(DateTime fromDate, DateTime toDate); 13 [VB] Public Sub SelectRange(ByVal fromDate As DateTime, ByVal toDate As 14 DateTime) 15 [JScript] public function SelectRange(fromDate : DateTime, toDate : DateTime); 16 17 Description 18 Adds the specified range of dates to the 19 System.Web.UI.WebControls.SelectedDatesCollection . 20 Use this method to add the specified range of dates to the 21 System.Web.UI.WebControls.SelectedDatesCollection . A System.DateTime 22 object that specifies the initial date to add to the 23 System.Web.UI.WebControls.SelectedDatesCollection. A System.DateTime 24

1	object that specifies the end date to add to the
2	System.Web.UI.WebControls.SelectedDatesCollection.
3	ServerValidateEventArgs class (System.Web.UI.WebControls)
4	ToString
5	
6	
7	Description
8	Provides data for the
9	System.Web.UI.WebControls.CustomValidator.ServerValidate event of the
10	System.Web.UI.WebControls.CustomValidator control. This class cannot be
11	inherited.
12	A System.Web.UI.WebControls.ServerValidateEventArgs object is
13	passed to the System.Web.UI.WebControls.CustomValidator.ServerValidate
14	event handler to provide event data to the handler. The
15	System.Web.UI.WebControls.CustomValidator.ServerValidate event is raised
16	when validation is performed on the server. This allows you to perform a custom
17	server-side validation routine on the value of an input control (with a
18	System.Web.UI.WebControls.CustomValidator control associated with it) in
19	the event handler.
20	ServerValidateEventArgs
21	Example Syntax:
22	ToString
23	
24	[C#] public ServerValidateEventArgs(string value, bool isValid);
25	[C++] public: ServerValidateEventArgs(String* value, bool isValid);

1	[VB] Public Sub New(ByVal value As String, ByVal isValid As Boolean)
2	[JScript] public function ServerValidateEventArgs(value : String, isValid :
3	Boolean);
4	
5	Description
6	Initializes a new instance of the
7	System.Web.UI.WebControls.ServerValidateEventArgs class.
8	Use this constructor to create and initialize a new instance of the
9	System.Web.UI.WebControls.ServerValidateEventArgs class. The value to
10	validate. true to indicate that the value passes validation; otherwise, false.
11	IsValid
12	ToString
13	
14	[C#] public bool IsValid {get; set;}
15	[C++] public:property bool get_IsValid();public:property void
16	set_IsValid(bool);
17	[VB] Public Property IsValid As Boolean
18	[JScript] public function get IsValid(): Boolean;public function set
19	IsValid(Boolean);
20	
21	Description
22	Gets or sets whether the value specified by the
23	System.Web.UI.WebControls.ServerValidateEventArgs.Value property passes
24	validation.
25	

	1	Once your validation routine finishes, use the
	2	System.Web.UI.WebControls.ServerValidateEventArgs.IsValid property to
	3	indicate whether the value specified by the
	4	System.Web.UI.WebControls.ServerValidateEventArgs.Value property passes
	5	validation. This value determines whether input control associated with the
	6	System.Web.UI.WebControls.CustomValidator control passes validation.
	7	Value
	8	ToString
	9	
James	10	[C#] public string Value {get;}
	11	[C++] public:property String* get_Value();
	12	[VB] Public ReadOnly Property Value As String
	13	[JScript] public function get Value() : String;
	14	
	15	Description
	16	Gets the value to validate in the custom event handler for the
	17	System.Web.UI.WebControls.CustomValidator.ServerValidate event.
	18	Use the System.Web.UI.WebControls.ServerValidateEventArgs.Value
	19	property determine the value to validate in the custom event handler for the
	20	System.Web.UI.WebControls.CustomValidator.ServerValidate event. Notice
	21	that you cannot programmatically change this value.
	22	ServerValidateEventHandler delegate (System.Web.UI.WebControls)
	23	ToString

# Description

2

3

5

7

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Represents the method that will handle the

System.Web.UI.WebControls.CustomValidator.ServerValidate event of a System.Web.UI.WebControls.CustomValidator control. The source of the event. A System.Web.UI.WebControls.ServerValidateEventArgs that contains the event data.

Use this delegate to provide custom code that performs validation on the server. Your code needs to validate the

System.Web.UI.WebControls.ServerValidateEventArgs.Value property of the args parameter passed to the delegate. Store the result of the validation in the System.Web.UI.WebControls.ServerValidateEventArgs.IsValid property of the args parameter.

Style class (System.Web.UI.WebControls)

**ToString** 

#### Description

Represents attributes that define the style of a Web Forms control and provides methods to manage the style.

Style

Example Syntax:

**ToString** 

25

```
[C#] public Style();
    [C++] public: Style();
    [VB] Public Sub New()
    [JScript] public function Style(); Initializes a new instance of the
5
    System.Web.UI.WebControls.Style class without a specified state bag.
7
    Description
8
           Initializes a new instance of the System. Web. UI. WebControls. Style class.
           Style
10
           Example Syntax:
11
           ToString
12
13
    [C#] public Style(StateBag bag);
14
    [C++] public: Style(StateBag* bag);
15
    [VB] Public Sub New(ByVal bag As StateBag)
16
    [JScript] public function Style(bag: StateBag);
17
18
    Description
19
           Initializes a new instance of the System.Web.UI.WebControls.Style class
20
    with the specified state bag information. Contains the specified state bag
21
    information.
22
           BackColor
23
           ToString
24
```

```
1
    [C#] public Color BackColor {get; set;}
    [C++] public: property Color get BackColor(); public: property void
    set BackColor(Color);
    [VB] Public Property BackColor As Color
    [JScript] public function get BackColor(): Color; public function set
    BackColor(Color);
7
    Description
9
           Gets or sets the background color property of the
10
    System.Web.UI.WebControls.Style class.
11
           BorderColor
12
           ToString
13
14
    [C#] public Color BorderColor {get; set;}
15
    [C++] public: __property Color get_BorderColor();public: __property void
16
    set_BorderColor(Color);
17
    [VB] Public Property BorderColor As Color
18
    [JScript] public function get BorderColor(): Color; public function set
19
    BorderColor(Color);
20
21
    Description
22
           Gets or sets the border color property of the
23
    System.Web.UI.WebControls.Style class.
24
           BorderStyle
25
```

1	ToString
2	
3	[C#] public BorderStyle BorderStyle {get; set;}
4	[C++] public:property BorderStyle get_BorderStyle();public:property void
5	set_BorderStyle(BorderStyle);
6	[VB] Public Property BorderStyle As BorderStyle
7	[JScript] public function get BorderStyle(): BorderStyle; public function set
8	BorderStyle(BorderStyle);
9	
10	Description
11	Gets or sets the border style property of the
12	System.Web.UI.WebControls.Style class.
13	BorderWidth
14	ToString
15	
16	[C#] public Unit BorderWidth {get; set;}
17	[C++] public:property Unit get_BorderWidth();public:property void
18	set_BorderWidth(Unit);
19	[VB] Public Property BorderWidth As Unit
20	[JScript] public function get BorderWidth(): Unit;public function set
21	BorderWidth(Unit);
22	
23	Description
24	Gets or sets the border width property of the
25	System.Web.UI.WebControls.Style class.

Container CssClass **ToString** 3 5 Description 6 Gets or sets the CSS class property of the 7 System.Web.UI.WebControls.Style class. 8 DesignMode **Events** 10 Font 11 **ToString** 12 13 14 Description 15 Gets a System. Web. UI. Web Controls. Font Info object that contains the 16 font properties for the System. Web.UI. WebControls. Style class. 17 ForeColor 18 **ToString** 19 20 [C#] public Color ForeColor {get; set;} 21 [C++] public: \_\_property Color get\_ForeColor();public: \_\_property void set ForeColor(Color); 23 [VB] Public Property ForeColor As Color 24 [JScript] public function get ForeColor(): Color; public function set

Description

2

3

5

6

7

8

10

12

13

14

15

16

17

18

19

20

21

22

23

24

25

A protected property. Gets a value indicating whether any style elements have been defined in the state bag.

IsTrackingViewState

**ToString** 

[C#] protected bool IsTrackingViewState {get;}

[C++] protected: \_\_property bool get\_IsTrackingViewState();

[VB] Protected ReadOnly Property IsTrackingViewState As Boolean

[JScript] protected function get IsTrackingViewState(): Boolean;

Description

A protected method. Returns a value indicating whether any style elements have been defined in the state bag.

Return Value: true if there are style elements defined in the state bag; otherwise,

false.

Site

ViewState

**ToString** 

Description

Gets the state bag that holds the style elements.

The private field name for the state bag is case sensitive.

23

24

Width

**ToString** 

[C#] public Unit Width {get; set;}

[C++] public: \_\_property Unit get\_Width();public: \_\_property void

set\_Width(Unit);

[VB] Public Property Width As Unit

[JScript] public function get Width(): Unit; public function set Width(Unit);

Description

Gets or sets the width property of the **System.Web.UI.WebControls.Style** class.

AddAttributesToRender

[C#] public void AddAttributesToRender(HtmlTextWriter writer);

[C++] public: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Public Sub AddAttributesToRender(ByVal writer As HtmlTextWriter)

[JScript] public function AddAttributesToRender(writer: HtmlTextWriter);

Description

Adds all non-blank style attributes to the HTML output stream to be rendered to the client. The output stream that renders HTML content to the client.

AddAttributesToRender

[C#] public virtual void AddAttributesToRender(HtmlTextWriter writer,

	1	WebControl owner);
	2	[C++] public: virtual void AddAttributesToRender(HtmlTextWriter* writer,
	3	WebControl* owner);
	4	[VB] Overridable Public Sub AddAttributesToRender(ByVal writer As
	5	HtmlTextWriter, ByVal owner As WebControl)
	6	[JScript] public function AddAttributesToRender(writer: HtmlTextWriter, owner
	7	: WebControl);
	8	
	9	Description
	10	Adds all non-blank style attributes to the HTML output stream to be
	11	rendered to the client.
	12	CopyFrom
	13	
7	14	[C#] public virtual void CopyFrom(Style s);
	15	[C++] public: virtual void CopyFrom(Style* s);
	16	[VB] Overridable Public Sub CopyFrom(ByVal s As Style)
	17	[JScript] public function CopyFrom(s : Style);
	18	
	19	Description
	20	Copies non-blank elements from the specified style, overwriting existing
	21	style elements if necessary. The style to be copied.
	22	LoadViewState
	23	
	24	[C#] protected internal void LoadViewState(object state);
	25	[C++] protected public: void LoadViewState(Object* state);

MS1-863US.APP

[VB] Protected Friend Dim Sub LoadViewState(ByVal state As Object) [JScript] package function LoadViewState(state : Object); 2 3 Description 4 A protected method. Load the previously saved state. The previously saved 5 state. 6 MergeWith 7 8 [C#] public virtual void MergeWith(Style s); [C++] public: virtual void MergeWith(Style\* s); 10 [VB] Overridable Public Sub MergeWith(ByVal s As Style) 11 [JScript] public function MergeWith(s: Style); 13 Description 14 Copies non-blank elements from the specified style, but will not overwrite 15 any existing style elements. The style to be copied. 16 Reset 17 18 [C#] public virtual void Reset(); 19 [C++] public: virtual void Reset(); 20 [VB] Overridable Public Sub Reset() 21 [JScript] public function Reset(); 22 23 Description 24 Clears out any defined style elements from the state bag. 25

2

3

6

7

8

9

10

12

13

14

15

16

17

18

19

20

21

22

23

24

α .		C1 . 1 .
SOTIO	V 16V	vState
Save	AICA	Volute

[C#] protected internal virtual object SaveViewState();

[C++] protected public: virtual Object\* SaveViewState();

[VB] Overridable Protected Friend Dim Function SaveViewState() As Object

[JScript] package function SaveViewState(): Object;

Description

A protected method. Saves any state that has been modified after the System.Web.UI.WebControls.Style.TrackViewState method was invoked.

Return Value: An object that represents the saved state. The default is null.

SetBit

[C#] protected internal virtual void SetBit(int bit);

[C++] protected public: virtual void SetBit(int bit);

[VB] Overridable Protected Friend Dim Sub SetBit(ByVal bit As Integer)

[JScript] package function SetBit(bit: int);

Description

A protected internal method.

IStateManager.LoadViewState

[C#] void IStateManager.LoadViewState(object state);

[C++] void IStateManager::LoadViewState(Object\* state);

[VB] Sub LoadViewState(ByVal state As Object) Implements

1	IStateManager.LoadViewState
2	[JScript] function IStateManager.LoadViewState(state : Object);
3	IStateManager.SaveViewState
4	
5	[C#] object IStateManager.SaveViewState();
6	[C++] Object* IStateManager::SaveViewState();
7	[VB] Function SaveViewState() As Object Implements
8	IStateManager.SaveViewState
9	[JScript] function IStateManager.SaveViewState(): Object;
10	IStateManager.TrackViewState
11	
12	[C#] void IStateManager.TrackViewState();
13	[C++] void IStateManager::TrackViewState();
14	[VB] Sub TrackViewState() Implements IStateManager.TrackViewState
15	[JScript] function IStateManager.TrackViewState();
16	ToString
17	
18	[C#] public override string ToString();
19	[C++] public: String* ToString();
20	[VB] Overrides Public Function ToString() As String
21	[JScript] public override function ToString() : String;
22	
23	Description
24	Overrides the ToString method to return System.String.Empty.
25	Return Value: System.String.Empty for all cases.

2

3

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

#### **TrackViewState**

[C#] protected internal virtual void TrackViewState();

[C++] protected public: virtual void TrackViewState();

[VB] Overridable Protected Friend Dim Sub TrackViewState()

[JScript] package function TrackViewState();

Description

A protected method. Marks the beginning for tracking state changes on the control. Any changes made after "mark" will be tracked and saved as part of the control viewstate.

Table class (System.Web.UI.WebControls)

**TrackViewState** 

Description

Constructs a table and defines its properties.

The System.Web.UI.WebControls.Table class allows you to build an HTML table and specify its characteristics in a straightforward manner, using the same abstract object model for building any other Web controls. A table can be built at design time given some static contents, but the power of a System.Web.UI.WebControls.Table Web control is often realized when the table is built programmatically with dynamic contents. In addition, as for other Web controls, the same code to render a table yields appropriate output accordingly for both downleveling and upleveling browsers.

Table Example Syntax: 2 TrackViewState 3 4 [C#] public Table(); [C++] public: Table(); [VB] Public Sub New() [JScript] public function Table(); 8 9 Description 10 Initializes a new instance of the System. Web. UI. WebControls. Table 11 class. 12 AccessKey 13 Attributes 14 BackColor 15 BackImageUrl 16 TrackViewState 17 18 19 Description 20 Indicates the URL of the background image to display behind the table. The 21 image will be tiled if it is smaller than the table. 22 BorderColor 23 BorderStyle 24 BorderWidth

25

CellPadding **TrackViewState** 2 3 Description Gets or sets the distance (in pixels) between the border and the contents of 6 the table cell. 7 CellSpacing 8 **TrackViewState** 9 10 [C#] public virtual int CellSpacing {get; set;} 11 [C++] public: \_\_property virtual int get\_CellSpacing();public: \_\_property virtual 12 void set CellSpacing(int); 13 [VB] Overridable Public Property CellSpacing As Integer 14 [JScript] public function get CellSpacing(): int;public function set 15 CellSpacing(int); 16 17 Description 18 Gets or sets the distance (in pixels) between table cells. 19 ChildControlsCreated 20 ClientID 21 Context 22 Controls 23 ControlStyle 24 ControlStyleCreated

CssClass Enabled 2 EnableViewState 3 **Events** Font 5 ForeColor GridLines TrackViewState 8 9 10 Description Gets or sets the gridlines property of the 12  ${\bf System. Web. UI. WebControls. Table\ class.}$ 13 HasChildViewState 14 Height 15 HorizontalAlign 16 TrackViewState 17 18

# Description

19

20

21

22

23

24

25

Gets or sets the horizontal alignment of the table within the page.

ID

IsTrackingViewState

NamingContainer

Page

Parent Rows 2 TrackViewState 3 4 5 Description Gets the collection of rows within the table. 7 This property is only used when building tables programmatically. At 8 design time, the property is set by declaring TableRow Web controls. Site 10 Style 11 TabIndex 12 TagKey 13 TagName 14 TemplateSourceDirectory 15 ToolTip 16 UniqueID 17 ViewState 18 ViewStateIgnoresCase 19 Visible 20 Width 21 AddAttributesToRender 22 23 [C#] protected override void AddAttributesToRender(HtmlTextWriter writer); 24 [C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

1	[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As
2	HtmlTextWriter)
3	[JScript] protected override function AddAttributesToRender(writer:
4	HtmlTextWriter);
5	
6	Description
7	A protected method. Adds information about the border color and border
8	width HTML attributes to the list of attributes to render. The output stream that
9	renders HTML content to the client.
10	CreateControlCollection
11	
12	[C#] protected override ControlCollection CreateControlCollection();
13	[C++] protected: ControlCollection* CreateControlCollection();
14	[VB] Overrides Protected Function CreateControlCollection() As
15	ControlCollection
16	[JScript] protected override function CreateControlCollection():
17	ControlCollection;
18	
19	Description
20	
21	CreateControlStyle
22	
23	[C#] protected override Style CreateControlStyle();
24	
25	[VB] Overrides Protected Function CreateControlStyle() As Style

[JScript] protected override function CreateControlStyle(): Style; 2 Description 3 A protected method. Creates a table control style. 4 Return Value: A System.Web.UI.WebControls.Style that specifies the table 5 control style. 6 RenderContents 7 8 [C#] protected override void RenderContents(HtmlTextWriter writer); 9 [C++] protected: void RenderContents(HtmlTextWriter\* writer); 10 [VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter) 11 [JScript] protected override function RenderContents(writer: HtmlTextWriter); 12 13 Description 14 A protected method. The output stream that renders HTML content to the 15 client. 16 TableCell class (System.Web.UI.WebControls) 17 **TrackViewState** 18 19 20 Description 21 Represents a cell in a **System.Web.UI.WebControls.Table** control. 22 The System.Web.UI.WebControls.TableCell class represents a cell in a 23 System.Web.UI.WebControls.Table control. You can use the 24 25

```
System.Web.UI.WebControls.TableCell.Text property to specify or determine
    the contents of the cell.
2
           TableCell
3
          Example Syntax:
           TrackViewState
6
    [C#] public TableCell();
    [C++] public: TableCell();
    [VB] Public Sub New()
    [JScript] public function TableCell();
10
11
    Description
           Initializes a new instance of the System.Web.UI.WebControls.TableCell
13
    class.
14
           Use this constructor to create and initialize a new instance of the
15
    System.Web.UI.WebControls.TableCell class.
16
           AccessKey
17
           Attributes
18
           BackColor
19
           BorderColor
20
           BorderStyle
21
           BorderWidth
22
           ChildControlsCreated
23
           ClientID
24
           ColumnSpan
25
```

TrackViewState

2

3

4 ||

6

7

8

10

12

14 15

16

17 18

19

2021

22

23

2425

Description

Gets or sets the number of columns in the

System.Web.UI.WebControls.Table control that the cell spans.

Use the **System.Web.UI.WebControls.TableCell.ColumnSpan** property to specify or determine the number of columns in the rendered table that the cell spans. For example, if you set this property to **2**, the cell takes up two columns in the **System.Web.UI.WebControls.Table** control.

Context

Controls

ControlStyle

ControlStyleCreated

CssClass

Enabled

EnableViewState

**Events** 

Font

ForeColor

HasChildViewState

Height

HorizontalAlign

TrackViewState

Description

2

3

5

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the horizontal alignment of the contents in the cell.

Use the System. Web. UI. WebControls. Table Cell. Horizontal Align property to specify the horizontal alignment of the contents of the cell. The following table lists the possible values.

ID

IsTrackingViewState

NamingContainer

Page

Parent

RowSpan

**TrackViewState** 

Description

Gets or sets the number of rows in the

**System.Web.UI.WebControls.Table** control that the cell spans.

Use the System.Web.UI.WebControls.TableCell.RowSpan property to specify or determine the number of rows in the rendered table that the cell spans. For example, if you set this property to 2, the cell takes up two rows in the

System.Web.UI.WebControls.Table control.

Site

Style

14

15

16

17

18

19

21

22

23

25

TagName
TemplateSourceDirectory
Text
TrackViewState

Description
Gets or sets the text contents of the cell.
Use the System.Web.UI.WebControls.TableCell.Text property to specify or determine the text contents of the cell. This property is commonly used to

TabIndex

or determine the text contents of the cell. This property is commonly used to

programmatically update the contents of a cell.

ToolTip

UniqueID

VerticalAlign

TrackViewState

20 Description

Gets or sets the vertical alignment of the contents in the cell.

Use the **System.Web.UI.WebControls.TableCell.VerticalAlign** property to specify the vertical alignment of the contents of the cell. The following table lists the possible values.

ViewState

ViewStateIgnoresCase

Visible

Width

Wrap

TrackViewState

### Description

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets a value that indicates whether the content of the cell wrap in the cell.

Use the **System.Web.UI.WebControls.TableCell.Wrap** property to specify or determine whether the content of the cell wrap in the cell.

AddAttributesToRender

[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);

[C++] protected: void AddAttributesToRender(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As

HtmlTextWriter)

[JScript] protected override function AddAttributesToRender(writer:

HtmlTextWriter);

## Description

A protected method. Adds information about the column span and row span to the list of attributes to render. The output stream that renders HTML content to the client.

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

[C#] protected override void AddParsedSubObject(object obj);

[C++] protected: void AddParsedSubObject(Object\* obj);

[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)

[JScript] protected override function AddParsedSubObject(obj : Object);

CreateControlStyle

[C#] protected override Style CreateControlStyle();

[C++] protected: Style\* CreateControlStyle();

[VB] Overrides Protected Function CreateControlStyle() As Style

[JScript] protected override function CreateControlStyle(): Style;

Description

A protected method. Creates a table item control style.

Return Value: A System.Web.UI.WebControls.Style that specifies the table item control style.

RenderContents

[C#] protected override void RenderContents(HtmlTextWriter writer);

[C++] protected: void RenderContents(HtmlTextWriter\* writer);

[VB] Overrides Protected Sub RenderContents(ByVal writer As HtmlTextWriter)

[JScript] protected override function RenderContents(writer: HtmlTextWriter);

Description

1	A protected method. The output stream that renders HTML content to the
2	client.
3	TableCellCollection class (System.Web.UI.WebControls)
4	TrackViewState
5	
6	
7	Description
8	Encapsulates a collection of
9	System.Web.UI.WebControls.TableHeaderCell and
10	System.Web.UI.WebControls.TableCell objects that make up a row in a
11	System.Web.UI.WebControls.Table control. This class cannot be inherited.
12	Use this class to programmatically manage a collection of
13	System.Web.UI.WebControls.TableCell objects that make up a row in a
14	System.Web.UI.WebControls.Table control. This class is commonly used to add
15	or remove cells from a row in a System.Web.UI.WebControls.Table control.
16	Count
17	TrackViewState
18	
19	[C#] public int Count {get;}
20	[C++] public:property int get_Count();
21	[VB] Public ReadOnly Property Count As Integer
22	[JScript] public function get Count(): int;
23	
24	Description
25	

2

3

5

7

8

9

10

11

13

14

17

18

19

21

22

23

24

Gets the number of System.Web.UI.WebControls.TableCell objects in  $the \ System. Web. UI. Web Controls. Table Cell Collection \ .$ Use this property to determine the number of cells in the  ${\bf System. Web. UI. Web Controls. Table Cell Collection}\ .\ {\bf The}$ System.Web.UI.WebControls.TableCellCollection.Count property is often used when iterating through the collection to determine the upper bound. **IsReadOnly TrackViewState** [C#] public bool IsReadOnly {get;} [C++] public: \_\_property bool get\_IsReadOnly(); [VB] Public ReadOnly Property IsReadOnly As Boolean [JScript] public function get IsReadOnly(): Boolean; Description 15 Gets a value indicating whether the 16  ${\bf System. Web. UI. WebControls. Table Cell Collection}\ is\ read-only.$ This property always returns false to indicate that the System.Web.UI.WebControls.TableCellCollection can be written to in all cases. **IsSynchronized** 20

**TrackViewState** 

[C#] public bool IsSynchronized {get;}

[C++] public: \_\_property bool get\_IsSynchronized();

[VB] Public ReadOnly Property IsSynchronized As Boolean

[JScript] public function get IsSynchronized(): Boolean; 2 Description Gets a value indicating whether access to the System.Web.UI.WebControls.TableCellCollection is synchronized (threadsafe). This property is derived from System. Collections. I Collection and is overridden to always return false. Item **TrackViewState** 10 11 [C#] public TableCell this[int index] {get;} [C++] public: property TableCell\* get Item(int index); 13 [VB] Public Default ReadOnly Property Item(ByVal index As Integer) As 14 **TableCell** 15 [JScript] returnValue = TableCellCollectionObject.Item(index); 16 17 Description 18 Gets a System. Web. UI. Web Controls. Table Cell from the 19 System.Web.UI.WebControls.TableCellCollection at the specified index. 20 Use this indexer to get an individual 21 System.Web.UI.WebControls.TableCell in the 22 System.Web.UI.WebControls.TableCellCollection at the specified index using 23 simple array notation. An ordinal index value that specifies the 24 System.Web.UI.WebControls.TableCell to return.

SyncRoot **TrackViewState** 3 [C#] public object SyncRoot {get;} [C++] public: property Object\* get SyncRoot(); [VB] Public ReadOnly Property SyncRoot As Object [JScript] public function get SyncRoot(): Object; 8 Description Gets the object that can be used to synchronize access to the 10  $System. Web. UI. Web Controls. Table Cell Collection \ .$ 11 The object returned in this implementation is the 12 System.Web.UI.WebControls.TableCellCollection object. 13 Add 14 15 [C#] public int Add(TableCell cell); [C++] public: int Add(TableCell\* cell); 17 [VB] Public Function Add(ByVal cell As TableCell) As Integer 18 [JScript] public function Add(cell: TableCell): int; 19 20 Description 21 Appends the specified System. Web. UI. WebControls. Table Cell to the end 22 of the System. Web. UI. Web Controls. Table Cell Collection. 23 Use this method to add the specified 24 System.Web.UI.WebControls.TableCell to the end of a

1	System. Web.UI. WebControls. Table Cell Collection. The
2	System.Web.UI.WebControls.TableCell to add to the collection.
3	AddAt
4	
5	[C#] public void AddAt(int index, TableCell cell);
6	[C++] public: void AddAt(int index, TableCell* cell);
7	[VB] Public Sub AddAt(ByVal index As Integer, ByVal cell As TableCell)
8	[JScript] public function AddAt(index : int, cell : TableCell);
9	
10	Description
11	Adds the specified System.Web.UI.WebControls.TableCell to the
12	System.Web.UI.WebControls.TableCellCollection at the specified index
13	location.
14	Use this method to insert the specified
15	System.Web.UI.WebControls.TableCell in a
16	System.Web.UI.WebControls.TableCellCollection at the specified index. The
17	location in the System.Web.UI.WebControls.TableCellCollection at which to
18	add the System.Web.UI.WebControls.TableCell. The
19	System.Web.UI.WebControls.TableCell to add to the
20	System.Web.UI.WebControls.TableCellCollection.
21	AddRange
22	
23	[C#] public void AddRange(TableCell[] cells);
24	[C++] public: void AddRange(TableCell* cells[]);
25	[VB] Public Sub AddRange(ByVal cells() As TableCell)

```
[JScript] public function AddRange(cells : TableCell[]);
  2
     Description
  3
            Clear
  5
     [C#] public void Clear();
 7
     [C++] public: _sealed void Clear();
     [VB] NotOverridable Public Sub Clear()
     [JScript] public function Clear(); Removes all
 10
     System.Web.UI.WebControls.TableCell objects from the
 11
     System. Web. UI. Web Controls. Table Cell Collection \ .
 12
13
    Description
14
           Removes all System. Web. UI. WebControls. Table Cell objects from the
15
    System. Web. UI. Web Controls. Table Cell Collection \ .
16
           Use this method to remove all System. Web.UI. WebControls. Table Cell
17
    objects from the System. Web.UI. WebControls. Table Cell Collection and set the
18
    System.Web.UI.WebControls.TableCellCollection.Count property to 0.
19
           CopyTo
20
21
    [C#] public void CopyTo(Array array, int index);
22
    [C++] public: __sealed void CopyTo(Array* array, int index);
23
    [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
    Integer)
25
```

1	[JScript] public function CopyTo(array: Array, index: int);
2	
3	Description
4	Copies the items from the
5	System.Web.UI.WebControls.TableCellCollection to the specified
6	System.Array, starting with the specified index in the System.Array.
7	Use this method to copy the contents of the
8	System.Web.UI.WebControls.TableCellCollection into the specified
9	System.Array starting at the specified index. A zero-based System.Array that
10	receives the copied items from the
11	System.Web.UI.WebControls.TableCellCollection. The first index in the
12	specified System.Array to receive the items.
13	GetCellIndex
14	
15	[C#] public int GetCellIndex(TableCell cell);
16	[C++] public: int GetCellIndex(TableCell* cell);
17	[VB] Public Function GetCellIndex(ByVal cell As TableCell) As Integer
18	[JScript] public function GetCellIndex(cell: TableCell): int;
19	
20	Description
21	Returns a value that represents the index of the specified
22	System.Web.UI.WebControls.TableCell from the
23	System.Web.UI.WebControls.TableCellCollection .
24	Return Value: The index of the specified
25	System.Web.UI.WebControls.TableCell within the

System.Web.UI.WebControls.TableCellCollection . The default is -1, which indicates that a match has not been found. Use this method to determine the index of the specified 3 System.Web.UI.WebControls.TableCell in the System.Web.UI.WebControls.TableCellCollection . If the specified System.Web.UI.WebControls.TableCell is not found, an index of -1 is returned. The System.Web.UI.WebControls.TableCell to get the index of in the System.Web.UI.WebControls.TableCellCollection. GetEnumerator 9 10 [C#] public IEnumerator GetEnumerator(); 11 [C++] public: sealed IEnumerator\* GetEnumerator(); [VB] NotOverridable Public Function GetEnumerator() As IEnumerator 13 [JScript] public function GetEnumerator(): IEnumerator; 14 15 Description 16 Returns a System. Collections. IEnumerator that contains all 17 System.Web.UI.WebControls.TableCell objects in the 18 System.Web.UI.WebControls.TableCellCollection . 19 Return Value: A System.Collections.IEnumerator that contains all 20 System.Web.UI.WebControls.TableCell objects within the 21 System. Web. UI. Web Controls. Table Cell Collection. 22 Use this method to create a System. Collections. IE numerator that can be 23 iterated through easily to get each item in the 24  $System. Web. UI. Web Controls. Table Cell Collection \ .$ 25

ndr <u>.</u>	1
	1
12	13
	1.
	14
4	15
	16
	17
	18
	20
	21
	22

1	Remove
2	
3	[C#] public void Remove(TableCell cell);
4	[C++] public: void Remove(TableCell* cell);
5	[VB] Public Sub Remove(ByVal cell As TableCell)
6	[JScript] public function Remove(cell : TableCell);
7	
8	Description
9	Removes the specified System.Web.UI.WebControls.TableCell from the
10	System.Web.UI.WebControls.TableCellCollection .
11	Use this method to remove the specified
12	System.Web.UI.WebControls.TableCell from a
13	System.Web.UI.WebControls.TableCellCollection . The
14	System.Web.UI.WebControls.TableCell to remove from the
15	System.Web.UI.WebControls.TableCellCollection.
16	RemoveAt
17	
18	[C#] public void RemoveAt(int index);
19	[C++] public:sealed void RemoveAt(int index);
20	[VB] NotOverridable Public Sub RemoveAt(ByVal index As Integer)
21	[JScript] public function RemoveAt(index : int);
22	
23	Description
24	Removes a System.Web.UI.WebControls.TableCell from the
25	System.Web.UI.WebControls.TableCellCollection at the specified index.

2

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use this method to remove a System.Web.UI.WebControls.TableCell from a System. Web. UI. WebControls. Table Cell Collection at the specified index. The index of the System.Web.UI.WebControls.TableCell to remove from  $the \ System. Web. UI. Web Controls. Table Cell Collection.$ IList.Add [C#] int IList.Add(object o); [C++] int IList::Add(Object\* o); [VB] Function Add(ByVal o As Object) As Integer Implements IList.Add [JScript] function IList.Add(o: Object): int; **IList.Contains** [C#] bool IList.Contains(object o); [C++] bool IList::Contains(Object\* o); [VB] Function Contains(ByVal o As Object) As Boolean Implements **IList.Contains** [JScript] function IList.Contains(o: Object): Boolean; IList.IndexOf [C#] int IList.IndexOf(object o); [C++] int IList::IndexOf(Object\* o); [VB] Function IndexOf(ByVal o As Object) As Integer Implements IList.IndexOf [JScript] function IList.IndexOf(o: Object): int; IList.Insert

```
[C#] void IList.Insert(int index, object o);
     [C++] void IList::Insert(int index, Object* o);
 3
     [VB] Sub Insert(ByVal index As Integer, ByVal o As Object) Implements
     IList.Insert
     [JScript] function IList.Insert(index: int, o: Object);
            IList.Remove
 7
 8
     [C#] void IList.Remove(object o);
 9
     [C++] void IList::Remove(Object* o);
 10
     [VB] Sub Remove(ByVal o As Object) Implements IList.Remove
11
     [JScript] function IList.Remove(o: Object);
12
           TableCellControlBuilder class (System.Web.UI.WebControls)
13
           ToString
14
15
16
    Description
17
           Interacts with the parser to build a
18
    System. Web. UI. WebControls. Table Cell\ control.
19
           To create a custom control builder for a
20
    System.Web.UI.WebControls.TableCell derived control, you need to inherit
21
    from this class.
22
           TableCellControlBuilder
23
           Example Syntax:
24
           ToString
25
```

1	
2	[C#] public TableCellControlBuilder();
3	[C++] public: TableCellControlBuilder();
4	[VB] Public Sub New()
5	[JScript] public function TableCellControlBuilder();
6	ControlType
7	FChildrenAsProperties
8	FIsNonParserAccessor
9	HasAspCode
10	ID
11	InDesigner
12	NamingContainerType
13	Parser
14	TagName
15	AllowWhitespaceLiterals
16	
17	[C#] public override bool AllowWhitespaceLiterals();
18	[C++] public: bool AllowWhitespaceLiterals();
19	[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
20	[JScript] public override function AllowWhitespaceLiterals(): Boolean;
21	
22	Description
23	Specifies whether white space literals are allowed.
24	Return Value: false for all cases.
25	

1

This method overrides

System.Web.UI.ControlBuilder.AllowWhitespaceLiterals to ignore white space in the System.Web.UI.WebControls.TableCell control.

TableHeaderCell class (System.Web.UI.WebControls)

**ToString** 

Description

Represents a heading cell within a **System.Web.UI.WebControls.Table** control.

The System.Web.UI.WebControls.TableHeaderCell class represents a heading cell in a System.Web.UI.WebControls.Table control. You can use the System.Web.UI.WebControls.TableCell.Text property to specify or determine the contents of the heading cell.

**TableHeaderCell** 

Example Syntax:

**ToString** 

[C#] public TableHeaderCell();

[C++] public: TableHeaderCell();

[VB] Public Sub New()

[JScript] public function TableHeaderCell();

Description

23

24

25

 ${\bf System. Web. UI. WebControls. Table Header Cell\ class.}$ 2 Use this contructor to create and initialize a new instance of the 3  ${\bf System. Web. UI. WebControls. Table Header Cell\ class.}$ AccessKey 5 Attributes BackColor 7 BorderColor BorderStyle 9 BorderWidth 10 ChildControlsCreated 11 ClientID 12 ColumnSpan 13 Context 14 Controls 15 ControlStyle 16 ControlStyleCreated 17 CssClass 18 Enabled 19 EnableViewState 20 **Events** 21 **Font** 22 ForeColor 23

Initializes a new instance of the

HasChildViewState Height 2128

24

25

1	HorizontalAlign
2	ID
3	IsTrackingViewState
4	NamingContainer
5	Page
6	Parent
7	RowSpan
8	Site
9	Style
10	TabIndex
11	TagKey
12	TagName
13	TemplateSourceDirectory
14	Text
15	ToolTip
16	UniqueID
17	VerticalAlign
18	ViewState
19	ViewStateIgnoresCase
20	Visible
21	Width
22	Wrap
23	TableItemStyle class (System.Web.UI.WebControls)
24	TrackViewState
25	

Description

4	Specifies the style for an item in a table control.
5	The System.Web.UI.WebControls.TableItemStyle class represents the
6	style properties for an item in the System.Web.UI.WebControls.Table control.
7	You can control the a Specifies the style of the table item.
8	TableItemStyle
9	Example Syntax:
10	TrackViewState
11	
12	[C#] public TableItemStyle();
13	[C++] public: TableItemStyle();
14	[VB] Public Sub New()
15	[JScript] public function TableItemStyle(); Creates a new instance of the
16	System.Web.UI.WebControls.TableItemStyle class.
17	
18	Description
19	Creates a new instance of the
20	System.Web.UI.WebControls.TableItemStyle class.
21	TableItemStyle
22	Example Syntax:

TrackViewState

[C#] public TableItemStyle(StateBag bag);

[C++] public: TableItemStyle(StateBag\* bag); [VB] Public Sub New(ByVal bag As StateBag) [JScript] public function TableItemStyle(bag: StateBag); 3 4 Description 5 Creates a new instance of the 6  ${\bf System. Web. UI. WebControls. Table I tem Style\ class\ with\ the\ specified\ state\ bag.}$ 7 The specified state bag. 8 BackColor 9 BorderColor 10 BorderStyle 11 BorderWidth 12 Container 13 CssClass 14 DesignMode 15 **Events** 16 Font 17 ForeColor 18 Height 19 HorizontalAlign 20 **TrackViewState** 21 22 23 Description 24 Gets or sets the horizontal alignment of the cell content. 25

**IsEmpty** IsTrackingViewState 2 Site VerticalAlign TrackViewState 5 6 7 Description 8 Gets or sets the vertical alignment of the cell content. 9 ViewState 10 Width 11 Wrap 12 TrackViewState 13 14 15 Description 16 Gets or sets a value indicating whether the cell content wraps within the 17 cell. 18 AddAttributesToRender 19 20 [C#] public override void AddAttributesToRender(HtmlTextWriter writer, 21 WebControl owner); 22 [C++] public: void AddAttributesToRender(HtmlTextWriter\* writer, 23 WebControl\* owner); 24

[VB] Overrides Public Sub AddAttributesToRender(ByVal writer As

1	HtmlTextWriter, ByVal owner As WebControl)
2	[JScript] public override function AddAttributesToRender(writer:
3	HtmlTextWriter, owner: WebControl);
4	
5	Description
6	Adds information about horizontal alignment, vertical alignment, and wrap
7	to the list of attributes to render. The output stream that renders HTML content to
8	the client. The control that the style refers to.
9	CopyFrom
10	
11	[C#] public override void CopyFrom(Style s);
12	[C++] public: void CopyFrom(Style* s);
13	[VB] Overrides Public Sub CopyFrom(ByVal s As Style)
14	[JScript] public override function CopyFrom(s : Style);
15	
16	Description
17	Copies non-blank elements from the specified style, overwriting existing
18	style elements if necessary. The style to copy.
19	MergeWith
20	
21	[C#] public override void MergeWith(Style s);
22	[C++] public: void MergeWith(Style* s);
23	[VB] Overrides Public Sub MergeWith(ByVal s As Style)
24	[JScript] public override function MergeWith(s : Style);
25	

24

25

Description

3

Copies non-blank elements from the specified style, but will not overwrite any existing style elements. The style to copy.

Reset

[C#] public override void Reset();

[C++] public: void Reset();

[VB] Overrides Public Sub Reset()

[JScript] public override function Reset();

Description

Clears out any defined style elements from the state bag.

TableRow class (System.Web.UI.WebControls)

**TrackViewState** 

Description

Represents a row in a **System.Web.UI.WebControls.Table** control.

The **System.Web.UI.WebControls.TableRow** class represents a row in a **System.Web.UI.WebControls.Table** control.

**TableRow** 

Example Syntax:

TrackViewState

25

```
[C#] public TableRow();
   [C++] public: TableRow();
   [VB] Public Sub New()
   [JScript] public function TableRow();
5
6
   Description
          Initializes a new instance of the System.Web.UI.WebControls.TableRow
8
    class.
9
           Use this constructor to create and initialize a new instance of the
10
    {\bf System. Web. UI. WebControls. Table Row~class.}
11
           AccessKey
12
           Attributes
13
           BackColor
14
           BorderColor
15
           BorderStyle
16
           BorderWidth
17
           Cells
18
           TrackViewState
19
20
21
     Description
22
           Gets a collection of System. Web.UI. WebControls. Table Cell objects that
23
```

represent the cells of a row in a System. Web.UI. WebControls. Table control.

Use this property to programmatically control a collection of System.Web.UI.WebControls.TableCell objects that represent the cells from a row of the System. Web. UI. Web Controls. Table control. You can programmatically add, insert, or remove a System.Web.UI.WebControls.TableCell object from the collection. ChildControlsCreated ClientID Context Controls ControlStyle 10 ControlStyleCreated 11 CssClass Enabled 13 EnableViewState 14 **Events** 15 Font 16 ForeColor 17 HasChildViewState 18 Height 19 HorizontalAlign 20 **TrackViewState** 21 22 23

lee@hayes pik 509-324-9256 2136 MS1-863US.APP

Gets or sets the horizontal alignment of the contents in the row.

Description

24

Use the **System.Web.UI.WebControls.TableRow.HorizontalAlign** property to specify the horizontal alignment of the contents of the row. The following table lists the possible values.

ID

1

3

4

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

IsTrackingViewState

NamingContainer

Page

Parent

Site

Style

TabIndex

TagKey

TagName

TemplateSourceDirectory

ToolTip

UniqueID

VerticalAlign

TrackViewState

Description

Gets or sets the vertical alignment of the contents in the row.

Use the **System.Web.UI.WebControls.TableRow.VerticalAlign** property to specify the vertical alignment of the contents of the row. The following table lists the possible values.

MS1-863US.APP

1	ViewState
2	ViewStateIgnoresCase
3	Visible
4	Width
5	CreateControlCollection
6	
7	[C#] protected override ControlCollection CreateControlCollection();
8	[C++] protected: ControlCollection* CreateControlCollection();
9	[VB] Overrides Protected Function CreateControlCollection() As
10	ControlCollection
11	[JScript] protected override function CreateControlCollection():
12	ControlCollection;
13	
14	Description
15	
16	CreateControlStyle
17	
18	[C#] protected override Style CreateControlStyle();
19	[C++] protected: Style* CreateControlStyle();
20	[VB] Overrides Protected Function CreateControlStyle() As Style
21	[JScript] protected override function CreateControlStyle() : Style;
22	
23	Description
24	
25	

A protected method. Creates a table item control style.

Return Value: A System. Web.UI. WebControls. Style that specifies the table item control style.

TableRowCollection class (System.Web.UI.WebControls)

**TrackViewState** 

### Description

3

5

6

7

9

10

11

12

13

15

16

17

18

20

21

22

23

24

Encapsulates a collection of **System.Web.UI.WebControls.TableRow** objects that represent a single row in a **System.Web.UI.WebControls.Table** control. This class cannot be inherited.

Use this class to programmatically manage a collection of **System.Web.UI.WebControls.TableRow** objects. This class is commonly used to add or remove rows from a **System.Web.UI.WebControls.Table** control.

Count

**TrackViewState** 

[C#] public int Count {get;}

[C++] public: \_ property int get Count();

[VB] Public ReadOnly Property Count As Integer

[JScript] public function get Count(): int;

#### Description

Gets the number of **System.Web.UI.WebControls.TableRow** objects in the **System.Web.UI.WebControls.TableRowCollection** .

1 Use this property to determine the number of rows in the System.Web.UI.WebControls.TableRowCollection . The System.Web.UI.WebControls.TableRowCollection.Count property is often used when iterating through the collection to determine the upper bound. IsReadOnly 5 **TrackViewState** 6 7 [C#] public bool IsReadOnly {get;} 8 [C++] public: \_\_property bool get\_IsReadOnly(); [VB] Public ReadOnly Property IsReadOnly As Boolean 10 [JScript] public function get IsReadOnly(): Boolean; 11 12 Description 13 Gets a value indicating whether the 14 System. Web. UI. WebControls. Table Row Collection is read-only. 15 This property always returns false to indicate that the 16 System.Web.UI.WebControls.TableRowCollection can be written to in all 17 cases. 18 IsSynchronized 19 **TrackViewState** 20 21 [C#] public bool IsSynchronized {get;} [C++] public: property bool get IsSynchronized(); 23 [VB] Public ReadOnly Property IsSynchronized As Boolean [JScript] public function get IsSynchronized(): Boolean; 25

7	
The state of the s	
r.	
<u> </u>	
: P	
d dans tag	
5	
:	

SyncRoot

	•
1	
2	Description
3	Gets a value indicating whether access to the
4	System.Web.UI.WebControls.TableRowCollection is synchronized (thread-
5	safe).
6	This property is derived from System.Collections.ICollection and is
7	overridden to always return false.
8	Item
9	TrackViewState
10	
11	[C#] public TableRow this[int index] {get;}
12	[C++] public:property TableRow* get_Item(int index);
13	[VB] Public Default ReadOnly Property Item(ByVal index As Integer) As
14	TableRow
15	[JScript] returnValue = TableRowCollectionObject.Item(index);
16	
17	Description
18	Gets a System.Web.UI.WebControls.TableRow from the
19	System.Web.UI.WebControls.TableRowCollection at the specified index.
20	Use this indexer to get an individual
21	System.Web.UI.WebControls.TableRow from the
22	System.Web.UI.WebControls.TableRowCollection at the specified index using

simple array notation. An ordinal index value that specifies which

 ${\bf System. Web. UI. WebControls. Table Row\ to\ return.}$ 

1	TrackViewState
2	
3	[C#] public object SyncRoot {get;}
4	[C++] public:property Object* get_SyncRoot();
5	[VB] Public ReadOnly Property SyncRoot As Object
6	[JScript] public function get SyncRoot() : Object;
7	
8	Description
9	Gets the object that can be used to synchronize access to the
10	System.Web.UI.WebControls.TableRowCollection .
11	The object returned in this implementation is the
12	System.Web.UI.WebControls.TableRowCollection object itself.
13	Add
14	
15	[C#] public int Add(TableRow row);
16	[C++] public: int Add(TableRow* row);
17	[VB] Public Function Add(ByVal row As TableRow) As Integer
18	[JScript] public function Add(row : TableRow) : int;
19	
20	Description
21	Appends the specified System.Web.UI.WebControls.TableRow to the
22	end of the System.Web.UI.WebControls.TableRowCollection .
23	Use this method to add a System.Web.UI.WebControls.TableRow to the
24	end of a System.Web.UI.WebControls.TableRowCollection . The
25	

1	System. Web. UI. Web Controls. Table Row to add to the
2	System.Web.UI.WebControls.TableRowCollection.
3	AddAt
4	
5	[C#] public void AddAt(int index, TableRow row);
6	[C++] public: void AddAt(int index, TableRow* row);
7	[VB] Public Sub AddAt(ByVal index As Integer, ByVal row As TableRow)
8	[JScript] public function AddAt(index : int, row : TableRow);
9	
10	Description
11	Adds the specified System.Web.UI.WebControls.TableRow to the
12	System.Web.UI.WebControls.TableRowCollection at the specified index
13	location.
14	Use this method to insert the specified
15	System.Web.UI.WebControls.TableRow in a
16	System.Web.UI.WebControls.TableRowCollection at the specified index. The
17	location in the System.Web.UI.WebControls.TableRowCollection at which to
18	add the System.Web.UI.WebControls.TableRow. The
19	System.Web.UI.WebControls.TableRow to add to the
20	System.Web.UI.WebControls.TableRowCollection.
21	AddRange
22	
23	[C#] public void AddRange(TableRow[] rows);
24	[C++] public: void AddRange(TableRow* rows[]);
25	[VB] Public Sub AddRange(ByVal rows() As TableRow)

```
[JScript] public function AddRange(rows : TableRow[]);
 2
    Description
           Clear
 6
    [C#] public void Clear();
 7
    [C++] public: __sealed void Clear();
    [VB] NotOverridable Public Sub Clear()
    [JScript] public function Clear();
10
11
    Description
12
           Removes all System. Web.UI. WebControls. Table Row controls from the
13
    System.Web.UI.WebControls.TableRowCollection .
           Use this method to remove all System.Web.UI.WebControls.TableRow
15
    objects from the System. Web. UI. Web Controls. Table Row Collection and set the
16
    System.Web.UI.WebControls.TableRowCollection.Count property to 0.
17
           CopyTo
18
19
    [C#] public void CopyTo(Array array, int index);
20
    [C++] public: _ sealed void CopyTo(Array* array, int index);
21
    [VB] NotOverridable Public Sub CopyTo(ByVal array As Array, ByVal index As
    Integer)
23
    [JScript] public function CopyTo(array : Array, index : int);
25
```

-				
1)2	SCI	'nn	tic	11/
$\mathcal{L}$	$\omega \omega r$	$\iota \nu$	$\iota\iota\iota$	,,,

3

6

9

10

11

12

13

14

15

16

17

18

19

20

22

23

Copies the items from the

System.Web.UI.WebControls.TableRowCollection to the specified

System.Array, starting with the specified index in the System.Array.

Use this method to copy the contents of the

System.Web.UI.WebControls.TableRowCollection into the specified

System.Array, starting at the specified index. A zero-based System.Array that receives the copied items from the

**System.Web.UI.WebControls.TableRowCollection**. The first position in the specified **System.Array** to receive copied contents.

GetEnumerator

[C#] public IEnumerator GetEnumerator();

[C++] public: \_\_sealed IEnumerator\* GetEnumerator();

[VB] NotOverridable Public Function GetEnumerator() As IEnumerator

[JScript] public function GetEnumerator(): IEnumerator;

## Description

Returns an System. Collections. IEnumerator that contains all

System. Web. UI. Web Controls. Table Row objects within the

System.Web.UI.WebControls.TableRowCollection .

Return Value: A System.Collections.IEnumerator that contains all

System.Web.UI.WebControls.TableRow objects within the

 $System. Web. UI. Web Controls. Table Row Collection \ .$ 

1 Use this method to create a System.Collections.IEnumerator that can be iterated through easily to get each item in the System.Web.UI.WebControls.TableRowCollection . GetRowIndex 5 [C#] public int GetRowIndex(TableRow row); [C++] public: int GetRowIndex(TableRow\* row); [VB] Public Function GetRowIndex(ByVal row As TableRow) As Integer [JScript] public function GetRowIndex(row : TableRow) : int; 9 10 Description 11 Returns a value that represents the index of the specified 12 System.Web.UI.WebControls.TableRow from the 13 System.Web.UI.WebControls.TableRowCollection . 14 Return Value: The ordinal index position of the specified 15 System. Web. UI. Web Controls. Table Row within the collection. The default is -1, 16 which indicates that the specified System. Web. UI. Web Controls. Table Row has 17 not been found. 18 Use this method to determine the index of the specified 19 System.Web.UI.WebControls.TableRow in the 20 System.Web.UI.WebControls.TableRowCollection . If the specified 21 System. Web. UI. WebControls. Table Row is not found, an index of -1 is returned. 22 The System.Web.UI.WebControls.TableRow to search for in the 23 System.Web.UI.WebControls.TableRowCollection. Remove 25

```
[C#] public void Remove(TableRow row);
    [C++] public: void Remove(TableRow* row);
    [VB] Public Sub Remove(ByVal row As TableRow)
    [JScript] public function Remove(row : TableRow);
6
    Description
7
          Removes the specified System. Web.UI. WebControls. Table Row from the
8
    System.Web.UI.WebControls.TableRowCollection .
          Use this method to remove the specified
10
    System.Web.UI.WebControls.TableRow from a
11
    System.Web.UI.WebControls.TableRowCollection . The
12
    System.Web.UI.WebControls.TableRow to remove from the
13
    System.Web.UI.WebControls.TableRowCollection.
          RemoveAt
15
16
    [C#] public void RemoveAt(int index);
17
    [C++] public: sealed void RemoveAt(int index);
18
    [VB] NotOverridable Public Sub RemoveAt(ByVal index As Integer)
    [JScript] public function RemoveAt(index : int);
20
21
    Description
22
          Removes a System. Web.UI. WebControls. Table Row from the
23
    System.Web.UI.WebControls.TableRowCollection at the specified index.
24
25
```

2

3

5

6

7

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Use this method to remove a System. Web. UI. Web Controls. Table Row from the System. Web.UI. WebControls. Table RowCollection at the specified index. The index of the System.Web.UI.WebControls.TableRow to remove from the System. Web.UI. WebControls. Table RowCollection. IList.Add [C#] int IList.Add(object o); [C++] int IList::Add(Object\* o); [VB] Function Add(ByVal o As Object) As Integer Implements IList.Add [JScript] function IList.Add(o: Object): int; **IList.Contains** [C#] bool IList.Contains(object o); [C++] bool IList::Contains(Object\* o); [VB] Function Contains(ByVal o As Object) As Boolean Implements **IList.Contains** [JScript] function IList.Contains(o: Object): Boolean; IList.IndexOf [C#] int IList.IndexOf(object o); [C++] int IList::IndexOf(Object\* o); [VB] Function IndexOf(ByVal o As Object) As Integer Implements IList.IndexOf [JScript] function IList.IndexOf(o: Object): int; IList.Insert

```
[C#] void IList.Insert(int index, object o);
    [C++] void IList::Insert(int index, Object* o);
    [VB] Sub Insert(ByVal index As Integer, ByVal o As Object) Implements
    IList.Insert
    [JScript] function IList.Insert(index: int, o: Object);
           IList.Remove
7
8
    [C#] void IList.Remove(object o);
9
    [C++] void IList::Remove(Object* o);
10
    [VB] Sub Remove(ByVal o As Object) Implements IList.Remove
11
    [JScript] function IList.Remove(o: Object);
12
           TableStyle class (System.Web.UI.WebControls)
13
           ToString
14
15
16
    Description
17
           Specifies the style for a table control.
18
           This is used primarily by control developers.
19
           TableStyle
20
           Example Syntax:
21
           ToString
22
23
    [C#] public TableStyle();
    [C++] public: TableStyle();
```

```
[VB] Public Sub New()
    [JScript] public function TableStyle(); Initializes a new instance of the
    System.Web.UI.WebControls.TableStyle class.
4
    Description
5
          Initializes a new instance of the System.Web.UI.WebControls.TableStyle
6
    class using default values.
           TableStyle
8
           Example Syntax:
           ToString
10
11
    [C#] public TableStyle(StateBag bag);
    [C++] public: TableStyle(StateBag* bag);
13
    [VB] Public Sub New(ByVal bag As StateBag)
    [JScript] public function TableStyle(bag: StateBag);
15
16
    Description
17
           Initializes a new instance of the System.Web.UI.WebControls.TableStyle
18
    class with the specified state bag information.
           BackColor
20
           BackImageUrl
21
           ToString
22
23
24
    Description
```

1 | Gets or sets the URL of an image to display in the background of a table control. 2 The image will be tiled if it is smaller than the table. 3 BorderColor BorderStyle 5 BorderWidth CellPadding **ToString** 8 9 10 Description 11 Gets or sets the distance between the border and the contents of the table 12 cell. 13 CellSpacing 14 **ToString** 15 16 [C#] public virtual int CellSpacing {get; set;} 17 [C++] public: \_\_property virtual int get CellSpacing();public: \_\_property virtual 18 void set\_CellSpacing(int); 19 [VB] Overridable Public Property CellSpacing As Integer 20 [JScript] public function get CellSpacing(): int;public function set 21 CellSpacing(int); 22 23 Description 24 Gets or sets the distance between table cells. 25

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Container
CssClass
DesignMode
Events
Font
ForeColor
GridLines
ToString

## Description

Gets or sets the gridlines property of the table.

Height

HorizontalAlign

**ToString** 

## Description

Gets or sets the horizontal alignment of the table within the page.

**IsEmpty** 

IsTrackingViewState

Site

ViewState

Width

AddAttributesToRender

1	
2	[C#] public override void AddAttributesToRender(HtmlTextWriter writer,
3	WebControl owner);
4	[C++] public: void AddAttributesToRender(HtmlTextWriter* writer,
5	WebControl* owner);
6	[VB] Overrides Public Sub AddAttributesToRender(ByVal writer As
7	HtmlTextWriter, ByVal owner As WebControl)
8	[JScript] public override function AddAttributesToRender(writer:
9	HtmlTextWriter, owner: WebControl);
10	
11	Description
12	Adds information about the background image, callspacing, cellpadding,
13	gridlines, and alignment to the list of attributes to render. The output stream that
14	renders HTML content to the client. The control associated with the style.
15	CopyFrom
16	
17	[C#] public override void CopyFrom(Style s);
18	[C++] public: void CopyFrom(Style* s);
19	[VB] Overrides Public Sub CopyFrom(ByVal s As Style)
20	[JScript] public override function CopyFrom(s : Style);
21	

MergeWith

style elements if necessary. The style to copy.

Description

23

Copies non-blank elements from the specified style, overwriting existing

```
[C#] public override void MergeWith(Style s);
    [C++] public: void MergeWith(Style* s);
    [VB] Overrides Public Sub MergeWith(ByVal s As Style)
    [JScript] public override function MergeWith(s: Style);
 6
    Description
           Copies non-blank elements from the specified style, but will not overwrite
8
    any existing style elements. The style to copy.
 9
           Reset
10
11
    [C#] public override void Reset();
12
    [C++] public: void Reset();
13
    [VB] Overrides Public Sub Reset()
14
    [JScript] public override function Reset();
15
16
    Description
17
           Clears out any defined style elements from the state bag.
18
           TargetConverter class (System.Web.UI.WebControls)
19
           TrackViewState
20
21
22
    Description
23
           TargetConverter
24
           Example Syntax:
25
```

1	TrackViewState
2	
3	[C#] public TargetConverter();
4	[C++] public: TargetConverter();
5	[VB] Public Sub New()
6	[JScript] public function TargetConverter();
7	GetStandardValues
8	
9	[C#] public override StandardValuesCollection
10	GetStandardValues(ITypeDescriptorContext context);
11	[C++] public: StandardValuesCollection*
12	GetStandardValues(ITypeDescriptorContext* context);
13	[VB] Overrides Public Function GetStandardValues(ByVal context As
14	ITypeDescriptorContext) As StandardValuesCollection
15	[JScript] public override function GetStandardValues(context:
16	ITypeDescriptorContext): StandardValuesCollection;
17	
18	Description
19	GetStandardValuesExclusive
20	
21	[C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext
22	context);
23	[C++] public: bool GetStandardValuesExclusive(ITypeDescriptorContext*
24	context);
25	[VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As

1	ITypeDescriptorContext) As Boolean
2	[JScript] public override function GetStandardValuesExclusive(context:
3	ITypeDescriptorContext): Boolean;
4	
5	Description
6	GetStandardValuesSupported
7	
8	[C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext
9	context);
10	[C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext*
11	context);
12	[VB] Overrides Public Function GetStandardValuesSupported(ByVal context As
13	ITypeDescriptorContext) As Boolean
14	[JScript] public override function GetStandardValuesSupported(context:
15	ITypeDescriptorContext): Boolean;
16	
17	Description
18	TemplateColumn class (System.Web.UI.WebControls)
19	ToString
20	
21	
22	Description
23	A column type for the System.Web.UI.WebControls.DataGrid control
24	that allows you to customize the layout of controls in the column.
25	

1	Use the System.Web.UI.WebControls.TemplateColumn column type in
2	a System.Web.UI.WebControls.DataGrid control to create a column with a
3	customized control layout.
4	TemplateColumn
5	Example Syntax:
6	ToString
7	
8	[C#] public TemplateColumn();
9	[C++] public: TemplateColumn();
 10	[VB] Public Sub New()
11	[JScript] public function TemplateColumn();
12	
13	Description
14	Initializes a new instance of the
15	System.Web.UI.WebControls.TemplateColumn class.
16	Use this constructor to create and initializes a new instance of the
17	System.Web.UI.WebControls.TemplateColumn class.
18	DesignMode
19	EditItemTemplate
20	ToString
21	
22	·
23	Description
24	Gets or sets the template for this column for the item selected for editing in
25	the System.Web.III.WebControls.DataGrid control

The state of the s		
and the state that the state that the same than		

Use the
${\bf System. Web. UI. WebControls. Template Column. Edit I tem Template}\ property\ to$
control the contents of the item selected for editing in the column of the
System.Web.UI.WebControls.DataGrid control.
FooterStyle
FooterTemplate
ToString
Description
Gets or sets the template for the footer section of this column in the
System.Web.UI.WebControls.DataGrid control.
Use the System.Web.UI.WebControls.DataList.FooterTemplate
property to control the contents of the footer section.
FooterText
HeaderImageUrl
HeaderStyle
HeaderTemplate

Description

**ToString** 

Gets or sets the template for the heading section of this column in the System.Web.UI.WebControls.DataGrid control.

1 Use the System. Web. UI. Web Controls. Data List. Header Template property to control the contents of the heading section. The appearance of the 2 header section is controlled by the 3 System.Web.UI.WebControls.DataList.HeaderStyle property. HeaderText IsTrackingViewState ItemStyle ItemTemplate 8 **ToString** 9 10 11 Description 12 Gets or sets the template for the items in this column of the 13 System.Web.UI.WebControls.DataGrid control. 14 Use the System.Web.UI.WebControls.TemplateColumn.ItemTemplate 15 property to control the contents of the items in the 16 System.Web.UI.WebControls.DataList control. 17 Owner 18 SortExpression 19 ViewState 20 Visible 21 InitializeCell 22 23 [C#] public override void InitializeCell(TableCell cell, int columnIndex, 24 ListItemType itemType);

1	[C++] public: void InitializeCell(TableCell* cell, int columnIndex, ListItemType
2	itemType);
3	[VB] Overrides Public Sub InitializeCell(ByVal cell As TableCell, ByVal
4	columnIndex As Integer, ByVal itemType As ListItemType)
5	[JScript] public override function InitializeCell(cell: TableCell, columnIndex: int,
6	itemType : ListItemType);
7	
8	Description
9	TextAlign enumeration (System.Web.UI.WebControls)
10	TrackViewState
11	
12	
13	Description
14	Specifies whether the text associated with a checkbox or radio button
15	control appears to the left or to the right of the control.
16	The System.Web.UI.WebControls.TextAlign enumeration represents the
17	different text alignment options for checkbox and radio button controls.
18	TrackViewState
19	
20	[C#] public const TextAlign Left;
21	[C++] public: const TextAlign Left;
22	[VB] Public Const Left As TextAlign
23	[JScript] public var Left : TextAlign;
24	
25	Description

MS1-863US.APP

Text associated with a checkbox or radio button control appears to the left 1 of the control. 2 **TrackViewState** 3 [C#] public const TextAlign Right; [C++] public: const TextAlign Right; [VB] Public Const Right As TextAlign [JScript] public var Right: TextAlign; 8 Description 10 Text associated with a checkbox or radio button control appears to the right 11 of the control. 12 TextBox class (System.Web.UI.WebControls) 13 **ToString** 14 15 16 Description 17 Constructs a text box and defines its properties. 18 The System.Web.UI.WebControls.TextBox server control is an input 19 control that lets the user enter text. By default, the 20 System.Web.UI.WebControls.TextBox.TextMode of the text box is SingleLine 21 , but it can be modified to be MultiLine or Password . 22 TextBox 23 Example Syntax: 24 **ToString** 

```
[C#] public TextBox();
    [C++] public: TextBox();
    [VB] Public Sub New()
    [JScript] public function TextBox();
6
    Description
7
           Initializes a new instance of the System.Web.UI.WebControls.TextBox
    class.
           AccessKey
10
           Attributes
11
           AutoPostBack
12
           ToString
13
14
15
    Description
16
           Gets or sets a value indicating whether an automatic postback to the server
17
    will occur whenever the user changes the content of the text box.
18
           BackColor
19
           BorderColor
20
           BorderStyle
21
           BorderWidth
22
           ChildControlsCreated
23
           ClientID
24
           Columns
25
```

**ToString** 3 DescriptionGets or sets the display width of the text box in characters. Context Controls ControlStyle ControlStyleCreated CssClass 10 Enabled 11 EnableViewState 12 **Events** 13 Font 14 ForeColor 15 HasChildViewState 16 Height 17 ID 18 IsTrackingViewState 19 MaxLength 20 **ToString** 21 22 23 Description 24 25

Gets or sets the maximum number of characters allowed in the text box.

1	This property is not applicable when the
2	System.Web.UI.WebControls.TextBox.TextMode property is set to
3	TextBoxMode.MultiLine .
4	NamingContainer
5	Page
6	Parent
7	ReadOnly
8	ToString
9	
10	
11	Description
12	Whether the textbox is in read-only mode.
13	Rows
14	ToString
15	
16	[C#] public virtual int Rows {get; set;}
17	[C++] public:property virtual int get_Rows();public:property virtual void
18	set_Rows(int);
19	[VB] Overridable Public Property Rows As Integer
20	[JScript] public function get Rows(): int;public function set Rows(int);
21	
22	Description
23	Gets or sets the display height of a multiline text box.
24	
25	

This property is only applicable when the System.Web.UI.WebControls.TextBox.TextMode property is set to 2 TextBoxMode.MultiLine. 3 Site Style **TabIndex** TagKey **ToString** 10 Description11 A protected property. Gets the HTML tag for the text box control. 12 TagName 13 **TemplateSourceDirectory** 14 Text 15 **ToString** 16 17 18 Description 19 Gets or sets the text content of the text box. 20 TextMode 21 **ToString** 22 23 [C#] public virtual TextBoxMode TextMode {get; set;} 24  $[C++]\ public: \_\_property\ virtual\ TextBoxMode\ get\_TextMode(); public:$ 

	1	property virtual void set_TextMode(TextBoxMode);
	2	[VB] Overridable Public Property TextMode As TextBoxMode
	3	[JScript] public function get TextMode(): TextBoxMode;public function set
	4	TextMode(TextBoxMode);
	5	
	6	Description
	7	Gets or sets the behavior mode of the text box.
	8	ToolTip
	9	UniqueID
	10	ViewState
	11	ViewStateIgnoresCase
hard had but then had the the the	12	Visible
Smil Sun	13	Width
	14	Wrap
	15	ToString
	16	
	17	
	18	Description
	19	Gets or sets a value indicating whether the text content wraps within the
	20	text box.
	21	This property is only applicable when the
	22	System.Web.UI.WebControls.TextBox.TextMode property is set to
	23	TextBoxMode.MultiLine .
	24	ToString
	25	

1	
2	
3	Description
4	Occurs when the content of the text box is changed upon server postback.
5	AddAttributesToRender
6	
7	[C#] protected override void AddAttributesToRender(HtmlTextWriter writer);
8	[C++] protected: void AddAttributesToRender(HtmlTextWriter* writer);
9	[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As
10	HtmlTextWriter)
11	[JScript] protected override function AddAttributesToRender(writer:
12	HtmlTextWriter);
13	
14	Description
15	AddParsedSubObject
16	
17	[C#] protected override void AddParsedSubObject(object obj);
18	[C++] protected: void AddParsedSubObject(Object* obj);
19	[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)
20	[JScript] protected override function AddParsedSubObject(obj : Object);
21	
22	Description
23	Overridden to only allow literal controls to be added as Text property.
24	OnPreRender
25	

1	
2	[C#] protected override void OnPreRender(EventArgs e);
3	[C++] protected: void OnPreRender(EventArgs* e);
4	[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)
5	[JScript] protected override function OnPreRender(e: EventArgs);
6	
7	Description
8	
9	OnTextChanged
10	
11	[C#] protected virtual void OnTextChanged(EventArgs e);
12	[C++] protected: virtual void OnTextChanged(EventArgs* e);
13	[VB] Overridable Protected Sub OnTextChanged(ByVal e As EventArgs
14	[JScript] protected function OnTextChanged(e: EventArgs);
15	
16	Description
17	Raises the <b>TextChanged</b> event.
18	Render
19	
20	[C#] protected override void Render(HtmlTextWriter writer);
21	[C++] protected: void Render(HtmlTextWriter* writer);
22	[VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)
23	[JScript] protected override function Render(writer: HtmlTextWriter);
24	
25	Description

control.

1	IPostBackDataHandler.LoadPostData
2	
3	[C#] bool IPostBackDataHandler.LoadPostData(string postDataKey,
4	NameValueCollection postCollection);
5	[C++] bool IPostBackDataHandler::LoadPostData(String* postDataKey,
6	NameValueCollection* postCollection);
7	[VB] Function LoadPostData(ByVal postDataKey As String, ByVal
8	postCollection As NameValueCollection) As Boolean Implements
9	IPostBackDataHandler.LoadPostData
10	[JScript] function IPostBackDataHandler.LoadPostData(postDataKey : String,
11	postCollection : NameValueCollection) : Boolean;
12	IPostBackDataHandler.RaisePostDataChangedEvent
13	
14	[C#] void IPostBackDataHandler.RaisePostDataChangedEvent();
15	[C++] void IPostBackDataHandler::RaisePostDataChangedEvent();
16	[VB] Sub RaisePostDataChangedEvent() Implements
17	IPostBackDataHandler.RaisePostDataChangedEvent
18	[JScript] function IPostBackDataHandler.RaisePostDataChangedEvent();
19	TextBoxControlBuilder class (System.Web.UI.WebControls)
20	TrackViewState
21	
22	
23	Description
24	Interacts with the parser to build a System.Web.UI.WebControls.TextBox

1	To create a custom control builder for a
2	System.Web.UI.WebControls.TextBox derived control, you need to inherit from
3	this class.
4	TextBoxControlBuilder
5	Example Syntax:
6	TrackViewState
7	
8	[C#] public TextBoxControlBuilder();
9	[C++] public: TextBoxControlBuilder();
10	[VB] Public Sub New()
11	[JScript] public function TextBoxControlBuilder();
12	ControlType
13	FChildrenAsProperties
14	FIsNonParserAccessor
15	HasAspCode
16	ID
17	InDesigner
18	NamingContainerType
19	Parser
20	TagName
21	AllowWhitespaceLiterals
22	
23	[C#] public override bool AllowWhitespaceLiterals();
24	[C++] public: bool AllowWhitespaceLiterals();
25	[VB] Overrides Public Function AllowWhitespaceLiterals() As Boolean
	2   3   4   5   6   7   8   9   10   11   12   13   14   15   16   17   18   19   20   21   22   23   24

[JScript] public override function AllowWhitespaceLiterals(): Boolean; 2 Description 3 Specifies whether white space literals are allowed. Return Value: false for all cases. This method overrides 6 System.Web.UI.ControlBuilder.AllowWhitespaceLiterals to ignore white 7 space in the System. Web. UI. WebControls. TextBox control. 8 HtmlDecodeLiterals 9 10 [C#] public override bool HtmlDecodeLiterals(); 11 [C++] public: bool HtmlDecodeLiterals(); 12 [VB] Overrides Public Function HtmlDecodeLiterals() As Boolean 13 [JScript] public override function HtmlDecodeLiterals(): Boolean; 14 15 Description 16 17 TextBoxMode enumeration (System.Web.UI.WebControls) 18 **ToString** 19 20 21 Description 22 Specifies the behavior mode of the text box. 23 The System.Web.UI.WebControls.TextBoxMode enumeration represents 24 the different display options for System. Web.UI. WebControls. TextBox controls. 25

1	ToString
2	
3	[C#] public const TextBoxMode MultiLine;
4	[C++] public: const TextBoxMode MultiLine;
5	[VB] Public Const MultiLine As TextBoxMode
6	[JScript] public var MultiLine : TextBoxMode;
7	
8	Description
9	Multi line entry mode.
10	ToString
11	
12	[C#] public const TextBoxMode Password;
13	[C++] public: const TextBoxMode Password;
14	[VB] Public Const Password As TextBoxMode
15	[JScript] public var Password : TextBoxMode;
16	
17	Description
18	Password entry mode.
19	ToString
20	
21	[C#] public const TextBoxMode SingleLine;
22	[C++] public: const TextBoxMode SingleLine;
23	[VB] Public Const SingleLine As TextBoxMode

[JScript] public var SingleLine : TextBoxMode;

	1	
	2	Description
	3	Single-line entry mode.
	4	TitleFormat enumeration (System.Web.UI.WebControls)
	5	ToString
	6	
	7	
	8	Description
	9	Specifies the title format for the displayed month in the
Ì	10	System.Web.UI.WebControls.Calendar control.
	11	The System.Web.UI.WebControls.TitleFormat enumeration represents
	12	the different title formats for the System.Web.UI.WebControls.Calendar
	13	control.
	14	ToString
	15	
	16	[C#] public const TitleFormat Month;
	17	[C++] public: const TitleFormat Month;
	18	[VB] Public Const Month As TitleFormat
	19	[JScript] public var Month: TitleFormat;
	20	
	21	Description
	22	Title displayed with only the month but not the year. For example,
	23	"January".
	24	ToString
	25	

1 [C#] public const TitleFormat MonthYear; [C++] public: const TitleFormat MonthYear; 3 [VB] Public Const MonthYear As TitleFormat [JScript] public var MonthYear : TitleFormat; 5 6 Description 7 Title displayed with both the month and the year. For example, "January 8 2000". 9 Unit structure (System.Web.UI.WebControls) 10 **ToString** 11 12 13 Description 14 Defines the fields, properties, and methods of the 15 System.Web.UI.WebControls.Unit structure. 16 **ToString** 17 18 [C#] public static readonly Unit Empty; 19 [C++] public: static Unit Empty; 20 [VB] Public Shared ReadOnly Empty As Unit 21 [JScript] public static var Empty: Unit; 22 23 Description 24 Specifies an empty unit. This field is read-only.

1	Unit
2	Example Syntax:
3	ToString
4	
5	[C#] public Unit(double value);
6	[C++] public: Unit(double value);
7	[VB] Public Sub New(ByVal value As Double)
8	[JScript] public function Unit(value : double);
9	
10	Description
11	Initializes a new instance of the System.Web.UI.WebControls.Unit
12	structure with the specified double-precision floating point number as the unit
13	value and Pixel as the (default) unit type. Represents the specified unit value.
14	Unit
15	Example Syntax:
16	ToString
17	
18	[C#] public Unit(int value);
19	[C++] public: Unit(int value);
20	[VB] Public Sub New(ByVal value As Integer)
21	[JScript] public function Unit(value : int); Initializes a new instance of the
22	System.Web.UI.WebControls.Unit structure.
23	
24	Description
25	

ı

2

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

Initializes a new instance of the System.Web.UI.WebControls.Unit structure with the specified 32-bit signed integer as the unit value and Pixel as the (default) unit type. Represents the specified unit value. Unit Example Syntax: **ToString** [C#] public Unit(string value); [C++] public: Unit(String\* value); [VB] Public Sub New(ByVal value As String) [JScript] public function Unit(value : String); Description Initializes a new instance of the System. Web.UI. WebControls. Unit structure with the specified text string that contains the unit value and unit type. If the unit type is not specified, the default is Pixel. The specified text string that contains the unit value and unit type. Unit Example Syntax: **ToString** [C#] public Unit(double value, UnitType type); [C++] public: Unit(double value, UnitType type); [VB] Public Sub New(ByVal value As Double, ByVal type As UnitType)

[JScript] public function Unit(value : double, type : UnitType);

1 Description 2 Initializes a new instance of the System.Web.UI.WebControls.Unit 3 structure with the specified double-precision floating point number as the unit value and the specified System. Web.UI. WebControls. Unit Type as the unit type. 5 Represents the specified unit value. Represents the specified unit type. 6 Unit 7 Example Syntax: 8 **ToString** 9 10 [C#] public Unit(string value, CultureInfo culture); 11 [C++] public: Unit(String\* value, CultureInfo\* culture); 12 [VB] Public Sub New(ByVal value As String, ByVal culture As CultureInfo) 13 [JScript] public function Unit(value : String, culture : CultureInfo); 14 15 Description 16 17 **IsEmpty** 18 **ToString** 19 20 [C#] public bool IsEmpty {get;} 21 [C++] public: \_property bool get\_IsEmpty(); 22 [VB] Public ReadOnly Property IsEmpty As Boolean 23 [JScript] public function get IsEmpty(): Boolean; 24 25

1	
2	Description
3	Gets a value indicating whether the System.Web.UI.WebControls.Unit is
4	empty.
5	Туре
6	ToString
7	
8	[C#] public UnitType Type {get;}
9	[C++] public:property UnitType get_Type();
10	[VB] Public ReadOnly Property Type As UnitType
11	[JScript] public function get Type() : UnitType;
12	
13	Description
14	Gets or sets the type of the System.Web.UI.WebControls.Unit.
15	Value
16	ToString
17	
18	[C#] public double Value {get;}
19	[C++] public:property double get_Value();
20	[VB] Public ReadOnly Property Value As Double
21	[JScript] public function get Value() : double;
22	
23	Description
24	Gets the value of the System.Web.UI.WebControls.Unit.
25	Equals

11	
1	
2	[C#] public override bool Equals(object obj);
3	[C++] public: bool Equals(Object* obj);
4	[VB] Overrides Public Function Equals(ByVal obj As Object) As Boolean
5	[JScript] public override function Equals(obj : Object) : Boolean;
6	
7	Description
8	Compares this System.Web.UI.WebControls.Unit with the specified
9	object.
10	Return Value: true if this System.Web.UI.WebControls.Unit has the same value
11	and type as the specified object; otherwise, false. The specified object for
12	comparison.
13	GetHashCode
14	
15	[C#] public override int GetHashCode();
16	[C++] public: int GetHashCode();
17	[VB] Overrides Public Function GetHashCode() As Integer
18	[JScript] public override function GetHashCode(): int;
19	
20	Description
21	
22	op_Equality
23	
24	[C#] public static bool operator ===(Unit left, Unit right);
25	[C++] public: static bool op_Equality(Unit left, Unit right);

```
[VB] returnValue = Unit.op_Equality(left, right)
    [JScript] returnValue = left == right;
2
3
    Description
           Compares two units to find out if they have the same value and type.
    Return Value: true if both units have the same value and type; otherwise, false.
    One of the two units being compared. The other of the two units being compared.
7
           op Implicit
8
9
    [C#] public static implicit operator Unit(int n);
10
    [C++] public: static Unit op_Implicit(int n);
11
    [VB] returnValue = Unit.op Implicit(n)
12
    [JScript] return Value = n;
13
14
    Description
15
           Implicitly creates a System.Web.UI.WebControls.Unit of type Pixel from
16
    the specified 32-bit unsigned integer. The specified 32-bit unsigned integer for
17
    creating a Unit.
18
           op Inequality
19
20
    [C#] public static bool operator !=(Unit left, Unit right);
21
    [C++] public: static bool op Inequality(Unit left, Unit right);
22
    [VB] returnValue = Unit.op Inequality(left, right)
23
    [JScript] returnValue = left != right;
25
```

Description Compares two units to find out if they have different values and/or types. 3 Return Value: true if both units have different values or different types; otherwise, 4 false. One of the two units being compared. The other of the two units being 5 compared. Parse 8 [C#] public static Unit Parse(string s); 9 [C++] public: static Unit Parse(String\* s); 10 [VB] Public Shared Function Parse(ByVal s As String) As Unit 11 [JScript] public static function Parse(s: String): Unit; 12 13 Description 14 15 Parse 16 17 [C#] public static Unit Parse(string s, CultureInfo culture); 18 [C++] public: static Unit Parse(String\* s, CultureInfo\* culture); 19 [VB] Public Shared Function Parse(ByVal s As String, ByVal culture As 20 CultureInfo) As Unit 21 [JScript] public static function Parse(s : String, culture : CultureInfo) : Unit; 22 23 Description 24 25

1	Percentage
2	
3	[C#] public static Unit Percentage(double n);
4	[C++] public: static Unit Percentage(double n);
5	[VB] Public Shared Function Percentage(ByVal n As Double) As Unit
6	[JScript] public static function Percentage(n : double) : Unit;
7	
8	Description
9	Creates a System.Web.UI.WebControls.Unit of type Percentage from the
10	specified 32-bit signed integer.
11	Return Value: A System.Web.UI.WebControls.Unit that represents the unit
12	created. The specified double precision floating point number for creating a unit.
13	Pixel
14	
15	[C#] public static Unit Pixel(int n);
16	[C++] public: static Unit Pixel(int n);
17	[VB] Public Shared Function Pixel(ByVal n As Integer) As Unit
18	[JScript] public static function Pixel(n: int): Unit;
19	
20	Description
21	Creates a System.Web.UI.WebControls.Unit of type Pixel from the
22	specified 32-bit signed integer.
23	Return Value: A System.Web.UI.WebControls.Unit that represents the unit
24	created. The specified 32-bit signed integer for creating a unit.
25	Point

1	
2	[C#] public static Unit Point(int n);
3	[C++] public: static Unit Point(int n);
4	[VB] Public Shared Function Point(ByVal n As Integer) As Unit
5	[JScript] public static function Point(n: int): Unit;
6	
7	Description
8	Creates a System.Web.UI.WebControls.Unit of type Point from the
9	specified 32-bit signed integer.
10	Return Value: A System.Web.UI.WebControls.Unit that represents the unit
11	created. The specified 32-bit signed integer for creating a unit.
12	ToString
13	
14	[C#] public override string ToString();
15	[C++] public: String* ToString();
16	[VB] Overrides Public Function ToString() As String
17	[JScript] public override function ToString(): String;
18	
19	Description
20	Converts a System. Web. UI. Web Controls. Unit to a System. String.
21	Return Value: A System.String represents this
22	System.Web.UI.WebControls.Unit.
23	ToString
24	
25	[C#] public string ToString(CultureInfo culture);

```
[C++] public: String* ToString(CultureInfo* culture);
    [VB] Public Function ToString(ByVal culture As CultureInfo) As String
2
    [JScript] public function ToString(culture : CultureInfo) : String;
3
4
    Description
5
6
           UnitConverter class (System.Web.UI.WebControls)
           ToString
9
10
    Description
11
           Specifies an interface to be overridden to provide unit conversion services.
12
    The base unit converter class.
13
           UnitConverter
14
           Example Syntax:
15
           ToString
16
17
    [C#] public UnitConverter();
18
    [C++] public: UnitConverter();
19
    [VB] Public Sub New()
20
    [JScript] public function UnitConverter();
21
           CanConvertFrom
22
23
    [C#] public override bool CanConvertFrom(ITypeDescriptorContext context,
24
    Type sourceType);
```

1	[C++] public: bool CanConvertFrom(ITypeDescriptorContext* context, Type*
2	sourceType);
3	[VB] Overrides Public Function CanConvertFrom(ByVal context As
4	ITypeDescriptorContext, ByVal sourceType As Type) As Boolean
5	[JScript] public override function CanConvertFrom(context:
6	ITypeDescriptorContext, sourceType : Type) : Boolean;
7	
8	Description
9	Returns a value indicating whether the unit converter can convert from the
10	specified source type.
11	Return Value: true if the source type can be converted from; otherwise, false. An
12	System.ComponentModel.ITypeDescriptorContext that specifies the context of
13	the object to convert. The type of the source.
14	ConvertFrom
15	
16	[C#] public override object ConvertFrom(ITypeDescriptorContext context,
17	CultureInfo culture, object value);
18	[C++] public: Object* ConvertFrom(ITypeDescriptorContext* context,
19	CultureInfo* culture, Object* value);
20	[VB] Overrides Public Function ConvertFrom(ByVal context As
21	ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
22	As Object
23	[JScript] public override function ConvertFrom(context : ITypeDescriptorContext,
24	culture : CultureInfo, value : Object) : Object;
25	

~		. •	
Desci	rın	111	m
レヒいし	$\iota \nu$	$\iota\iota\upsilon$	,,,

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Performs type conversion from the specified context, object and argument list.

Return Value: The object resulting from conversion. An

System.ComponentModel.ITypeDescriptorContext that indicates the context of the object to convert. A System.Globalization.CultureInfo object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in null for this parameter. The object to convert.

#### ConvertTo

 $[C\#]\ public\ override\ object\ Convert To (IType Descriptor Context\ context,$ 

CultureInfo culture, object value, Type destinationType);

[C++] public: Object\* ConvertTo(ITypeDescriptorContext\* context, CultureInfo\* culture, Object\* value, Type\* destinationType);

[VB] Overrides Public Function ConvertTo(ByVal context As

ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object,

ByVal destinationType As Type) As Object

[JScript] public override function ConvertTo(context: ITypeDescriptorContext,

culture: CultureInfo, value: Object, destinationType: Type): Object;

#### Description

Performs type conversion to the specified destination type given the specified context, object and argument list.

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Return Value: The object resulting from conversion. An

System.ComponentModel.ITypeDescriptorContext that indicates the context of the object to convert. A System.Globalization.CultureInfo object that represents information about a culture such as language, calendar system, and so on. This parameter is not used in this method. It is reserved for future versions of this method. You can optionally pass in null for this parameter. The object to convert. The type to convert to.

UnitType enumeration (System.Web.UI.WebControls)
ToString

### Description

Specifies the unit of measurement.

The **System.Web.UI.WebControls.UnitType** enumeration represents the different supported measurement units. Measurements can be represented in pixels, points, picas, inches, millimeters, centimeters, percentages, em, or ex.

**ToString** 

[C#] public const UnitType Cm;

[C++] public: const UnitType Cm;

[VB] Public Const Cm As UnitType

[JScript] public var Cm : UnitType;

# Description

Measurement is in centimeters.

1	ToString
2	
3	[C#] public const UnitType Em;
4	[C++] public: const UnitType Em;
5	[VB] Public Const Em As UnitType
6	[JScript] public var Em : UnitType;
7	
8	Description
9	Measurement is relative to the height of the parent element's font.
10	ToString
11	
12	[C#] public const UnitType Ex;
13	[C++] public: const UnitType Ex;
14	[VB] Public Const Ex As UnitType
15	[JScript] public var Ex : UnitType;
16	
17	Description
18	Measurement is relative to the height of the lowercase letter x of the parent
19	element's font.
20	ToString
21	
22	[C#] public const UnitType Inch;
23	[C++] public: const UnitType Inch;
24	[VB] Public Const Inch As UnitType
25	[JScript] public var Inch : UnitType;

	11	
	1	
	2	Description
	3	Measurement is in inches.
	4	ToString
	5	
	6	[C#] public const UnitType Mm;
	7	[C++] public: const UnitType Mm;
	8	[VB] Public Const Mm As UnitType
	9	[JScript] public var Mm : UnitType;
thereis thereis thereis	10	
	11	Description
	12	Measurement is in millimeters.
	13	ToString
Alde Hill II and alle II	14	
	15	[C#] public const UnitType Percentage;
	16	[C++] public: const UnitType Percentage;
	17	[VB] Public Const Percentage As UnitType
	18	[JScript] public var Percentage : UnitType;
	19	
	20	Description
	21	Measurement is a percentage relative to the parent element.
	22	ToString
	23	
	24	[C#] public const UnitType Pica;
	25	[C++] public: const UnitType Pica;
		n

	1	[VB] Public Const Pica As UnitType
	2	[JScript] public var Pica : UnitType;
	3	
	4	Description
	5	Measurement is in picas. A pica represents 12 points.
	6	ToString
	7	
	8	[C#] public const UnitType Pixel;
	9	[C++] public: const UnitType Pixel;
	10	[VB] Public Const Pixel As UnitType
9	11	[JScript] public var Pixel : UnitType;
	12	
	13	Description
	14	Measurement is in pixels.
	15	ToString
	16	
;	17	[C#] public const UnitType Point;
	18	[C++] public: const UnitType Point;
	19	[VB] Public Const Point As UnitType
	20	[JScript] public var Point : UnitType;
	21	
	22	Description
	23	Measurement is in points. A point represents 1/72 of an inch.
	24	ValidatedControlConverter class (System.Web.UI.WebControls)
	25	ToString

> 17 18

16

19

2021

22

24 25 Description

Shows a list of validatable controls in the Properties window.

GetStandardValues

[C#] public override StandardValuesCollection

GetStandardValues(ITypeDescriptorContext context);

[C++] public: StandardValuesCollection\*

GetStandardValues(ITypeDescriptorContext\* context);

[VB] Overrides Public Function GetStandardValues(ByVal context As

ITypeDescriptorContext) As StandardValuesCollection

[JScript] public override function GetStandardValues(context:

ITypeDescriptorContext): StandardValuesCollection;

Description

Returns a collection of standard values retrieved from the context specified by the specified type descriptor.

Return Value: A Standard Values Collection that represents the standard values collected from the specified context. A type descriptor that specifies the location of the context to convert from.

GetStandardValuesExclusive

[C#] public override bool GetStandardValuesExclusive(ITypeDescriptorContext context);

1	[C++] public: bool GetStandard values Exclusive(11 ypeDescriptorContext*
2	context);
3	[VB] Overrides Public Function GetStandardValuesExclusive(ByVal context As
4	ITypeDescriptorContext) As Boolean
5	[JScript] public override function GetStandardValuesExclusive(context:
6	ITypeDescriptorContext): Boolean;
7	
8	Description
9	Gets whether or not the context specified contains exclusive standard
10	values.
11	Return Value: true if the specified context contains exclusive standard values,
12	otherwise false. A type descriptor that indicates the context to convert from.
13	GetStandardValuesSupported
14	
15	[C#] public override bool GetStandardValuesSupported(ITypeDescriptorContext
16	context);
17	[C++] public: bool GetStandardValuesSupported(ITypeDescriptorContext*
18	context);
19	[VB] Overrides Public Function GetStandardValuesSupported(ByVal context As
20	ITypeDescriptorContext) As Boolean
21	[JScript] public override function GetStandardValuesSupported(context:
22	ITypeDescriptorContext): Boolean;
23	
24	Description
25	

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

Gets whether or not the specified context contains supported standard values.

Return Value: true if the specified context conatins supported standard values, otherwise false. A type descriptor that indicates the context to convert from.

ValidationCompareOperator enumeration (System.Web.UI.WebControls)
ToString

## Description

Specifies the validation comparison operators used by the System.Web.UI.WebControls.CompareValidator control.

The System.Web.UI.WebControls.ValidationCompareOperator enumeration represents the comparison operations that can be performed by the System.Web.UI.WebControls.CompareValidator control.

**ToString** 

[C#] public const ValidationCompareOperator DataTypeCheck;

[C++] public: const ValidationCompareOperator DataTypeCheck;

[VB] Public Const DataTypeCheck As ValidationCompareOperator

[JScript] public var DataTypeCheck : ValidationCompareOperator;

# Description

A comparison for data type only.

ToString

[C#] public const ValidationCompareOperator Equal;
[C++] public: const ValidationCompareOperator Equal;
[VB] Public Const Equal As ValidationCompareOperator
[JScript] public var Equal : ValidationCompareOperator;
Description
A comparison for equality.
ToString
[C4] muhlis const Volidation Compare Operator Greater Than:
[C#] public const ValidationCompareOperator GreaterThan;
[C++] public: const ValidationCompareOperator GreaterThan;
[VB] Public Const GreaterThan As ValidationCompareOperator
[JScript] public var GreaterThan : ValidationCompareOperator;
Description
A comparison for greater than.
ToString
[C#] public const ValidationCompareOperator GreaterThanEqual;
[C++] public: const ValidationCompareOperator GreaterThanEqual;
[VB] Public Const GreaterThanEqual As ValidationCompareOperator
[JScript] public var GreaterThanEqual : ValidationCompareOperator;
Description

1	A comparison for greater than or equal to.
2	ToString
3	
4	[C#] public const ValidationCompareOperator LessThan;
5	[C++] public: const ValidationCompareOperator LessThan;
6	[VB] Public Const LessThan As ValidationCompareOperator
7	[JScript] public var LessThan: ValidationCompareOperator;
8	
9	Description
10	A comparison for less than.
11	ToString
12	
13	[C#] public const ValidationCompareOperator LessThanEqual;
14	[C++] public: const ValidationCompareOperator LessThanEqual;
15	[VB] Public Const LessThanEqual As ValidationCompareOperator
16	[JScript] public var LessThanEqual : ValidationCompareOperator;
17	
18	Description
19	A comparison for less than or equal to.
20	ToString
21	
22	[C#] public const ValidationCompareOperator NotEqual;
23	[C++] public: const ValidationCompareOperator NotEqual;
24	[VB] Public Const NotEqual As ValidationCompareOperator
2.5	[IScript] public var NotEqual: ValidationCompareOperator;

1	
2	Description
3	A comparison for inequality.
4	ValidationDataType enumeration (System.Web.UI.WebControls)
5	ToString
6	
7	*
8	Description
9	Specifies the validation data types used by the
10	System.Web.UI.WebControls.CompareValidator and
11	System.Web.UI.WebControls.RangeValidator controls.
12	The System.Web.UI.WebControls.ValidationDataType enumeration
13	represents the different data types that the
14	System.Web.UI.WebControls.CompareValidator and
15	System.Web.UI.WebControls.RangeValidator controls can validate.
16	ToString
17	
18	[C#] public const ValidationDataType Currency;
19	[C++] public: const ValidationDataType Currency;
20	[VB] Public Const Currency As ValidationDataType
21	[JScript] public var Currency : ValidationDataType;
22	
23	Description
24	A currency data type.
25	ToString

1	
2	[C#] public const ValidationDataType Date;
3	[C++] public: const ValidationDataType Date;
4	[VB] Public Const Date As ValidationDataType
5	[JScript] public var Date : ValidationDataType;
6	
7	Description
8	A date data type.
9	ToString
10	
11	[C#] public const ValidationDataType Double;
12	[C++] public: const ValidationDataType Double;
13	[VB] Public Const Double As ValidationDataType
14	[JScript] public var Double : ValidationDataType;
15	
16	Description
17	A double data type.
18	ToString
19	
20	[C#] public const ValidationDataType Integer;
21	[C++] public: const ValidationDataType Integer;
22	[VB] Public Const Integer As ValidationDataType
23	[JScript] public var Integer : ValidationDataType;
24	
25	Description

MS1-863US.APP lee@hayes plic 509-324-9256

24

25

An integer data type. **ToString** 2 3 [C#] public const ValidationDataType String; [C++] public: const ValidationDataType String; 5 [VB] Public Const String As ValidationDataType [JScript] public var String : ValidationDataType; 7 8 Description 9 A string data type. 10 ValidationSummary class (System.Web.UI.WebControls) 11 **ToString** 12 13 14 Description 15 Displays a summary of all validation errors inline on a Web page, in a 16 message box, or both. 17 The System.Web.UI.WebControls.ValidationSummary class is used to 18 summarize the error messages from all validators on a Web page, in a single 19 location. The summary can be displayed as a list, as a bulleted list, or as a single 20 paragraph based on the 21 22

 $System. Web. UI. Web Controls. Validation Summary. Display Mode\ property.$ 

Example Syntax:

**ValidationSummary** 

**ToString** 

```
[C#] public ValidationSummary();
   [C++] public: ValidationSummary();
   [VB] Public Sub New()
    [JScript] public function ValidationSummary();
5
6
    Description
          Initializes a new instance of the
8
    System. Web. UI. Web Controls. Validation Summary \ class.
           The following table shows the initial property value for an instance of
10
    System. Web. UI. Web Controls. Validation Summary \ .
11
           AccessKey
12
           Attributes
13
           BackColor
14
           BorderColor
15
           BorderStyle
16
           BorderWidth
17
           ChildControlsCreated
18
           ClientID
19
           Context
20
           Controls
21
           ControlStyle
22
           ControlStyleCreated
23
           CssClass
24
           DisplayMode
```

ToString

# Description

2

3

5

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

Gets or sets the display mode of the validation summary.

Use this property to specify the display format of a

**System.Web.UI.WebControls.ValidationSummary** control. The summary may be displayed as a list, as a bulleted list, or as a single paragraph.

EnableClientScript

**ToString** 

[C#] public bool EnableClientScript {get; set;}

[C++] public: \_\_property bool get\_EnableClientScript();public: \_\_property void set EnableClientScript(bool);

[VB] Public Property EnableClientScript As Boolean

[JScript] public function get EnableClientScript() : Boolean; public function set EnableClientScript(Boolean);

# Description

Gets or sets a value indicating whether client-side validation is attempted on the browser.

Use this property to enable or disable client-side validation on the browser. When set to **true**, client-side validation is performed on the browser if the browser supports that feature. When set to **false**, no client-side validation is attempted, the **System.Web.UI.WebControls.ValidationSummary** control only

	1	updates itself on round-trips to the server, and the
	2	$System. Web. UI. Web Controls. Validation Summary. Show Message Box\ property$
	3	has no effect.
	4	Enabled
	5	EnableViewState
	6	Events
	7	Font
	8	ForeColor
	9	ToString
	10	
	11	
ing that had han had had had that	12	Description
	13	Gets or sets the fore color of the control.
	14	Use this property to specify the color that error messages from validation
* 4.15 4.17 1.14 1.14	15	controls are displayed.
	16	HasChildViewState
ilg Sig	17	HeaderText
	18	ToString
	19	
	20	
	21	Description
	22	Gets or sets the header text displayed at the top of the summary.
	23	Use this property to display a title for the
	24	System.Web.UI.WebControls.ValidationSummary control.
	25	Height

ID
IsTrackingViewState
NamingContainer
Page
Parent
ShowMessageBox
ToString

### Description

Gets or sets a value indicating whether the validation summary is displayed in a message box.

This property can be used in addition to the System.Web.UI.WebControls.ValidationSummary.ShowSummary property to control where the validation summary is displayed. If this property and System.Web.UI.WebControls.ValidationSummary.EnableClientScript are both set to true, the validation summary is displayed in a message box. If System.Web.UI.WebControls.ValidationSummary.EnableClientScript is set to false, this property has no effect.

ShowSummary

**ToString** 

[C#] public bool ShowSummary {get; set;}

[C++] public: \_\_property bool get\_ShowSummary();public: \_\_property void set ShowSummary(bool);

	1	[VB] Public Property ShowSummary As Boolean
	2	[JScript] public function get ShowSummary(): Boolean; public function set
	3	ShowSummary(Boolean);
	4	
	5	Description
	6	Gets or sets a value indicating whether the validation summary is displayed
	7	inline.
	8	This property can be used in addition to the
that that	9	System.Web.UI.WebControls.ValidationSummary.ShowMessageBox property
	10	to control where the validation summary is displayed. If this property is set to true
Դուժ հուն հում մոու վույի կոլժ կոլի Աոր	11	, the validation summary is displayed on the Web page.
	12	Site
in the second	13	Style
il dani mat li lime that	14	TabIndex
	15	TagKey
	16	TagName
	17	TemplateSourceDirectory
	18	ToolTip
	19	UniqueID
	20	ViewState
	21	ViewStateIgnoresCase
	22	Visible
	23	Width
	24	AddAttributesToRender
	25	

1	
2	[C#] protected override void AddAttributesToRender(HtmlTextWriter writer)
3	[C++] protected: void AddAttributesToRender(HtmlTextWriter* writer);
4	[VB] Overrides Protected Sub AddAttributesToRender(ByVal writer As
5	HtmlTextWriter)
6	[JScript] protected override function AddAttributesToRender(writer:
7	HtmlTextWriter);
8	
9	Description
10	AddAttributesToRender method.
11	OnPreRender
12	
13	[C#] protected override void OnPreRender(EventArgs e);
14	[C++] protected: void OnPreRender(EventArgs* e);
15	[VB] Overrides Protected Sub OnPreRender(ByVal e As EventArgs)
16	[JScript] protected override function OnPreRender(e : EventArgs);
17	
18	Description
19	PreRender method.
20	Render
21	
22	[C#] protected override void Render(HtmlTextWriter writer);
23	[C++] protected: void Render(HtmlTextWriter* writer);
24	[VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter)
25	[JScript] protected override function Render(writer : HtmlTextWriter);

	140.0	
:		
:	THE STREET WITH STREET STREET	
;	ľ	
;	::::::::::::::::::::::::::::::::::::::	
:	===	
•	Ц	
:	ij	
:	77	
:	127.	
:	127	
•	### P	
	# W. T.	
	:::::12	
:	in in it in	
	ısığ	
	1	
	::::7	

1	
2	Description
3	Render method.
4	ValidationSummaryDisplayMode enumeration
5	(System.Web.UI.WebControls)
6	TrackViewState
7	
8	
9	Description
10	Specifies the validation summary display mode used by the
11	System.Web.UI.WebControls.ValidationSummary control.
12	The System.Web.UI.WebControls.ValidationSummaryDisplayMode
13	enumeration represents the different display formats of a
14	System.Web.UI.WebControls.ValidationSummary control.
15	TrackViewState
16	
17	[C#] public const ValidationSummaryDisplayMode BulletList;
18	[C++] public: const ValidationSummaryDisplayMode BulletList;
19	[VB] Public Const BulletList As ValidationSummaryDisplayMode
20	[JScript] public var BulletList : ValidationSummaryDisplayMode;
21	
22	Description
23	Validation summary displayed in a bulleted list.
24	TrackViewState

1 [C#] public const ValidationSummaryDisplayMode List; [C++] public: const ValidationSummaryDisplayMode List; 3 [VB] Public Const List As ValidationSummaryDisplayMode [JScript] public var List : ValidationSummaryDisplayMode; 5 6 Description 7 Validation summary displayed in a list. 8 **TrackViewState** 9 10 [C#] public const ValidationSummaryDisplayMode SingleParagraph; [C++] public: const ValidationSummaryDisplayMode SingleParagraph; 12 [VB] Public Const SingleParagraph As ValidationSummaryDisplayMode 13 [JScript] public var SingleParagraph : ValidationSummaryDisplayMode; 14 15 Description 16 Validation summary displayed in a single paragraph. 17 ValidatorDisplay enumeration (System.Web.UI.WebControls) 18 **ToString** 19 20 21 Description 22 Specifies the display behavior of error messages in validation controls. 23  $The \ {\bf Validator Display System. Web. UI. Web Controls \ enumeration}$ 24 represents the different display behaviors of error messages in validation controls. 25

1	ToString
2	
3	[C#] public const ValidatorDisplay Dynamic;
4	[C++] public: const ValidatorDisplay Dynamic;
5	[VB] Public Const Dynamic As ValidatorDisplay
6	[JScript] public var Dynamic : ValidatorDisplay;
7	
8	Description
9	Validator content dynamically added to the page when validation fails.
10	ToString
11	
12	[C#] public const ValidatorDisplay None;
13	[C++] public: const ValidatorDisplay None;
14	[VB] Public Const None As ValidatorDisplay
15	[JScript] public var None : ValidatorDisplay;
16	
17	Description
18	Validator content never displayed inline.
19	ToString
20	
21	[C#] public const ValidatorDisplay Static;
22	[C++] public: const ValidatorDisplay Static;
23	
24	[JScript] public var Static : ValidatorDisplay;

11	
1	
2	Description
3	Validator content physically part of the page layout.
4	VerticalAlign enumeration (System.Web.UI.WebControls)
5	ToString
6	
7	
8	Description
9	Specifies the vertical alignment of an object or text in a control.
10	The System.Web.UI.WebControls.VerticalAlign enumeration represents
11	the different vertical alignment options for an object or text in a control.
12	ToString
13	
14	[C#] public const VerticalAlign Bottom;
15	[C++] public: const VerticalAlign Bottom;
16	[VB] Public Const Bottom As VerticalAlign
17	[JScript] public var Bottom : VerticalAlign;
18	
19	Description
20	Text or object is aligned with the bottom of the enclosing control.
21	ToString
22	
23	[C#] public const VerticalAlign Middle;
24	[C++] public: const VerticalAlign Middle;
25	[VB] Public Const Middle As VerticalAlign

[JScript] public var Middle: VerticalAlign; 2 Description 3 Text or object is aligned with the center of the enclosing control. **ToString** 5 6 [C#] public const VerticalAlign NotSet; [C++] public: const VerticalAlign NotSet; [VB] Public Const NotSet As VerticalAlign [JScript] public var NotSet : VerticalAlign; 10 11 Description 12 Vertical alignment is not set. 13 **ToString** 14 15 [C#] public const VerticalAlign Top; 16 [C++] public: const VerticalAlign Top; 17 [VB] Public Const Top As VerticalAlign 18 [JScript] public var Top : VerticalAlign; 19 20 Description 21 Text or object is aligned with the top of the enclosing control. 22 WebColorConverter class (System.Web.UI.WebControls) 23 **ToString** 24 25

1.	
1	
2	
3	Description
4	WebColorConverter
5	Example Syntax:
6	ToString
7	
8	[C#] public WebColorConverter();
9	[C++] public: WebColorConverter();
10	[VB] Public Sub New()
11	[JScript] public function WebColorConverter();
12	ConvertFrom
13	
14	[C#] public override object ConvertFrom(ITypeDescriptorContext context,
15	CultureInfo culture, object value);
16	[C++] public: Object* ConvertFrom(ITypeDescriptorContext* context,
17	CultureInfo* culture, Object* value);
18	[VB] Overrides Public Function ConvertFrom(ByVal context As
19	ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object)
20	As Object
21	[JScript] public override function ConvertFrom(context : ITypeDescriptorContext
22	culture : CultureInfo, value : Object) : Object;
23	
24	Description
25	ConvertTo

[C#] public override object ConvertTo(ITypeDescriptorContext context,
CultureInfo culture, object value, Type destinationType);
[C++] public: Object* ConvertTo(ITypeDescriptorContext* context, CultureInfo
culture, Object* value, Type* destinationType);
[VB] Overrides Public Function ConvertTo(ByVal context As
ITypeDescriptorContext, ByVal culture As CultureInfo, ByVal value As Object,
ByVal destinationType As Type) As Object
[JScript] public override function ConvertTo(context : ITypeDescriptorContext,
culture : CultureInfo, value : Object, destinationType : Type) : Object;
Description
WebControl class (System.Web.UI.WebControls)
ToString

# Description

Serves as the base class that defines the methods, properties and events common to all controls in the **System.Web.UI.WebControls** namespace.

Certain properties of the base control may not render on downlevel browsers for some or all controls. For example, the

System.Web.UI.WebControls.WebControl.AccessKey property will not render on downlevel browsers for any controls. See specific property for more details.

WebControl

Example Syntax:

1	ToString
2	
3	[C#] protected WebControl();
4	[C++] protected: WebControl();
5	[VB] Protected Sub New()
6	[JScript] protected function WebControl(); Initializes a new instance of the
7	System.Web.UI.WebControls.WebControl class.
8	
9	Description
10	Initializes a new instance of the
11	System.Web.UI.WebControls.WebControl class with a Span HTML tag.
12	WebControl
13	Example Syntax:
14	ToString
15	
16	[C#] public WebControl(HtmlTextWriterTag tag);
17	[C++] public: WebControl(HtmlTextWriterTag tag);
18	[VB] Public Sub New(ByVal tag As HtmlTextWriterTag)
19	[JScript] public function WebControl(tag: HtmlTextWriterTag);
20	
21	Description
22	Initializes a new instance of the
23	System.Web.UI.WebControls.WebControl class using the specified HTML tag.
24	One of the System.Web.UI.HtmlTextWriteTag values.
25	WebControl

1	Example Syntax:
2	ToString
3	
4	[C#] protected WebControl(string tag);
5	[C++] protected: WebControl(String* tag);
6	[VB] Protected Sub New(ByVal tag As String)
7	[JScript] protected function WebControl(tag: String);
8	
9	Description
10	Initializes a new instance of the
11	System.Web.UI.WebControls.WebControl class with the specified HTML tag.
12	An HTML tag.
13	AccessKey
14	ToString
15	
16	[C#] public virtual string AccessKey {get; set;}
17	[C++] public:property virtual String* get_AccessKey();public:property
18	virtual void set_AccessKey(String*);
19	[VB] Overridable Public Property AccessKey As String
20	[JScript] public function get AccessKey(): String; public function set
21	AccessKey(String);
22	
23	
24	Gets or sets the keyboard shortcut key (AccessKey) for setting focus to the
25	Web control.

2

3

4

5

6

7

9

Description

This property will not render on downlevel browsers for any controls. It is not HTML 4.0 and will only work in IE 4 or higher. Attributes **ToString** [C#] public AttributeCollection Attributes {get;} [C++] public: \_\_property AttributeCollection\* get\_Attributes(); [VB] Public ReadOnly Property Attributes As AttributeCollection 8 [JScript] public function get Attributes(): AttributeCollection; 10 Description 11 Gets the collection of arbitrary attributes (for rendering only) that do not 12 correspond to properties on the control. 13 This property will render on downlevel browsers for all controls. BackColor 15 **ToString** 16 17 [C#] public virtual Color BackColor {get; set;} [C++] public: \_\_property virtual Color get\_BackColor();public: \_\_property virtual 19 void set BackColor(Color); 20 [VB] Overridable Public Property BackColor As Color 21 [JScript] public function get BackColor(): Color; public function set 22 BackColor(Color); 23 24

1	Gets or sets the background color of the Web control.
2	This property will render on downlevel browsers for some controls only.
3	For example, System.Web.UI.WebControls.Table,
4	$System. Web. UI. Web Controls. Panel\ ,\ System. Web. UI. Web Controls. Data Grid$
5	, System.Web.UI.WebControls.Calendar , and
6	System.Web.UI.WebControls.ValidationSummary . It will also work for
7	System.Web.UI.WebControls.CheckBoxList,
8	System.Web.UI.WebControls.RadioButtonList and
9	System.Web.UI.WebControls.DataList if their RepeatLayout property is Table
10	and not Flow.
11	BorderColor
12	ToString
13	
14	[C#] public virtual Color BorderColor {get; set;}
15	[C++] public:property virtual Color get_BorderColor();public:property
16	virtual void set_BorderColor(Color);
17	[VB] Overridable Public Property BorderColor As Color
18	[JScript] public function get BorderColor() : Color; public function set
19	BorderColor(Color);
20	
21	Description
22	Gets or sets the border color of the Web control.
23	The property will render downlevel only for the same table-based controls
24	like the System.Web.UI.WebControls.WebControl.BackColor property.
25	However, it is output as the "bordercolor" attribute which is not part of the HTML

	1	3.2 standard. It works for Navigaor 4 and higher, and IE 3 and higher, but not most
	2	other browsers.
	3	BorderStyle
	4	ToString
	5	
	6	[C#] public virtual BorderStyle BorderStyle {get; set;}
	7	[C++] public:property virtual BorderStyle get_BorderStyle();public:
	8	property virtual void set_BorderStyle(BorderStyle);
	9	[VB] Overridable Public Property BorderStyle As BorderStyle
111111	10	[JScript] public function get BorderStyle() : BorderStyle; public function set
	11	BorderStyle(BorderStyle);
ff finns that	12	
Ant Toji haf han bal lag Lag Lag	13	Description
	14	Gets or sets the border style of the Web control.
the state of	15	This property will not render on downlevel browsers for any controls.
,	16	BorderWidth
; ;	17	ToString
	18	
	19	[C#] public virtual Unit BorderWidth {get; set;}
	20	[C++] public:property virtual Unit get_BorderWidth();public:property
	21	virtual void set_BorderWidth(Unit);
	22	[VB] Overridable Public Property BorderWidth As Unit
	23	[JScript] public function get BorderWidth(): Unit; public function set
	24	BorderWidth(Unit);
	25	

Description 2 Gets or sets the border width of the Web control. 3  $Use the {\bf \ System. Web. UI. WebControls. WebControl. Border Width}$ property to specify a border width for a control. 5 ChildControlsCreated 6 ClientID 7 Context 8 Controls 9 ControlStyle 10 **ToString** 11 12 13 Description 14 Gets the style of the Web control. This property is primarily used by control 15 developers. 16 ControlStyleCreated 17 **ToString** 18 19 [C#] public bool ControlStyleCreated {get;} 20 [C++] public: \_\_property bool get\_ControlStyleCreated(); 21 [VB] Public ReadOnly Property ControlStyleCreated As Boolean 22

Description

23

24

[JScript] public function get ControlStyleCreated(): Boolean;

1	CssClass
2	ToString
3	
4	[C#] public virtual string CssClass {get; set;}
5	[C++] public:property virtual String* get_CssClass();public:property virtual
6	<pre>void set_CssClass(String*);</pre>
7	[VB] Overridable Public Property CssClass As String
8	[JScript] public function get CssClass(): String; public function set
9	CssClass(String);
10	
11	Description
12	Gets or sets the CSS class rendered by the Web control.
13	This property will render on downlevel browsers for all controls.
14	Enabled
15	ToString
16	
17	[C#] public virtual bool Enabled {get; set;}
18	[C++] public:property virtual bool get_Enabled();public:property virtual
19	void set_Enabled(bool);
20	[VB] Overridable Public Property Enabled As Boolean
21	[JScript] public function get Enabled(): Boolean; public function set
22	Enabled(Boolean);
23	
24	Description
25	Gets or sets a value indicating whether the Web control is enabled.

This property will render on downlevel browsers for some controls only. **EnableViewState Events** Font **ToString** 5 Description 8 Gets font information of the Web control. 9 This property includes subproperties that can be accessed declaratively in 10 the form of "Font-Bold" or programmatically in the form of "Font.Bold". 11 ForeColor 12 **ToString** 13 14 [C#] public virtual Color ForeColor {get; set;} 15 [C++] public: \_\_property virtual Color get\_ForeColor();public: \_\_property virtual 16 void set ForeColor(Color); 17 [VB] Overridable Public Property ForeColor As Color 18 [JScript] public function get ForeColor(): Color; public function set 19 ForeColor(Color); 20 21 Description 22 Gets or sets the foreground color (typically the color of the text) of the Web 23 control. 24 25

1	This property will render on downlevel browsers for almost all controls,
2	except the System.Web.UI.WebControls.Image,
3	System.Web.UI.WebControls.AdRotator,
4	System.Web.UI.WebControls.HyperLink and
5	System.Web.UI.WebControls.LinkButton controls. In addition, it will be
6	rendered as tags on downlevel browsers.
7	HasChildViewState
8	Height
9	ToString
10	
11	
12	Description
13	Gets or sets the height of the Web control.
14	This property will render on downlevel browsers for some controls only. It
15	will not render downlevel for System.Web.UI.WebControls.Label,
16	System.Web.UI.WebControls.HyperLink,
17	System.Web.UI.WebControls.LinkButton, any validator controls, or for
18	System.Web.UI.WebControls.CheckBoxList,
19	System.Web.UI.WebControls.RadioButtonList and
20	System.Web.UI.WebControls.DataList when their RepeatLayout property is
21	Flow . Furthermore, only unit types of Pixel and percentage will work.
22	ID
23	IsTrackingViewState
24	NamingContainer
25	Page

Parent 1 Site 2 Style 3 **ToString** 5 6 Description Gets a collection of text attributes that will be rendered as a style attribute 8 on the outer tag of the Web control. 9 This property will render on downlevel browsers for all controls. 10 TabIndex 11 **ToString** 12 13 [C#] public virtual short TabIndex {get; set;} 14 [C++] public: \_\_property virtual short get\_TabIndex();public: \_\_property virtual 15 void set TabIndex(short); 16 [VB] Overridable Public Property TabIndex As Short 17 [JScript] public function get TabIndex(): Int16;public function set 18 TabIndex(Int16); 19 20 Description 21 Gets or sets the tab index of the Web control. 22 Use the System.Web.UI.WebControls.WebControl.TabIndex property 23 to specify or determine the tab index of a control on the Web Page. 24 TagKey 25

1	ToString
2	
3	[C#] protected virtual HtmlTextWriterTag TagKey {get;}
4	[C++] protected:property virtual HtmlTextWriterTag get_TagKey();
5	[VB] Overridable Protected ReadOnly Property TagKey As HtmlTextWriterTag
6	[JScript] protected function get TagKey(): HtmlTextWriterTag;
7	
8	Description
9	TagName
10	ToString
11	
12	[C#] protected virtual string TagName {get;}
13	[C++] protected:property virtual String* get_TagName();
14	[VB] Overridable Protected ReadOnly Property TagName As String
15	[JScript] protected function get TagName(): String;
16	
17	Description
18	A protected property. Gets the name of the control tag. This property is
19	primarily used by control developers.
20	TemplateSourceDirectory
21	ToolTip
22	ToString
23	
24	
25	Description

3

5

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Gets or sets the tool tip for the Web control to be displayed when the mouse cursor is over the control.

This property will not render on downlevel browsers for any controls.

UniqueID

ViewState

ViewStateIgnoresCase

Visible

Width

**ToString** 

## Description

Gets or sets the width of the Web control.

This property will render on downlevel browsers for some controls only. It will not render downlevel for **System.Web.UI.WebControls.Label**,

System.Web.UI.WebControls.HyperLink,

 ${\bf System. Web. UI. WebControls. Link Button}\ , \ {\bf any}\ validator\ controls, \ or\ for$ 

System.Web.UI.WebControls.CheckBoxList,

 $System. Web. UI. Web Controls. Radio Button List\ and$ 

 ${\bf System. Web. UI. WebControls. Data List} \ \ {\bf when} \ \ {\bf their} \ {\bf Repeat Layout} \ property \ is$ 

Flow . Furthermore, only unit types of Pixel and Percentage will work.

AddAttributesToRender

 $[C\#]\ protected\ virtual\ void\ AddAttributes ToRender (Html TextWriter\ writer);$ 

 $[C++]\ protected:\ virtual\ void\ AddAttributesToRender(HtmlTextWriter*\ writer);$ 

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

25

[VB] Overridable Protected Sub AddAttributesToRender(ByVal writer As HtmlTextWriter) [JScript] protected function AddAttributesToRender(writer: HtmlTextWriter); Description Adds to the specified writer those HTML attributes and styles that need to be rendered. This method is primarily used by control developers. The output stream that renders HTML content to the client. ApplyStyle [C#] public void ApplyStyle(Style s); [C++] public: void ApplyStyle(Style\* s); [VB] Public Sub ApplyStyle(ByVal s As Style) [JScript] public function ApplyStyle(s: Style); Description Copies any non-blank elements of the specified style to the Web control,

overwriting any existing style elements of the control. This method is primarily used by control developers. The style to be copied.

# CopyBaseAttributes

[C#] public void CopyBaseAttributes(WebControl controlSrc); [C++] public: void CopyBaseAttributes(WebControl\* controlSrc); [VB] Public Sub CopyBaseAttributes(ByVal controlSrc As WebControl) [JScript] public function CopyBaseAttributes(controlSrc : WebControl);

Description

1

2

3

5

8

10

11

12

13

15

16

17

18

19

20

21

22

23

24

25

 $Copies\ the\ {\bf System. Web. UI. WebControls. WebControl. Access Key}\ ,$ 

 $System. Web. UI. WebControls. WebControl. Enabled\ ,$ 

 $System. Web. UI. Web Controls. Web Control. Tool Tip\ ,\\$ 

 $System. Web. UI. WebControls. WebControl. TabIndex\ , \ {\tt and}$ 

System.Web.UI.WebControls.WebControl.Attributes properties onto the Web control from the specified source control. The source control with properties to be copied onto the Web control.

CreateControlStyle

[C#] protected virtual Style CreateControlStyle();

[C++] protected: virtual Style\* CreateControlStyle();

[VB] Overridable Protected Function CreateControlStyle() As Style

[JScript] protected function CreateControlStyle(): Style;

Description

A protected method. Creates the style object that is used internally to implement all style-related properties. Controls may override to create an appropriately typed style.

Return Value: A System.Web.UI.WebControls.Style that is used to implement all style-related properties of the control.

LoadViewState

[C#] protected override void LoadViewState(object savedState);

[C++] protected: void LoadViewState(Object\* savedState); 1 [VB] Overrides Protected Sub LoadViewState(ByVal savedState As Object) 2 [JScript] protected override function LoadViewState(savedState : Object); 3 4 Description 5 Loads previously saved state. Overridden to handle ViewState, Style, and 6 Attributes. Previously saved state. MergeStyle 8 9 [C#] public void MergeStyle(Style s); 10 [C++] public: void MergeStyle(Style\* s); 11 [VB] Public Sub MergeStyle(ByVal s As Style) 12 [JScript] public function MergeStyle(s: Style); 13 14 Description 15 Copies any non-blank elements of the specified style to the Web control, 16 but will not overwrite any existing style elements of the control. This method is primarily used by control developers. The style to be copied. 18 Render 19 20 [C#] protected override void Render(HtmlTextWriter writer); 21 [C++] protected: void Render(HtmlTextWriter\* writer); 22 [VB] Overrides Protected Sub Render(ByVal writer As HtmlTextWriter) 23 [JScript] protected override function Render(writer: HtmlTextWriter); 24 25

Description

2

3

4

5

6

7

9

10

11

12

13

15

16

17

18

19

20

21

22

23

24

Renders the control into the specified writer. The output stream that renders HTML content to the client.

RenderBeginTag

[C#] public virtual void RenderBeginTag(HtmlTextWriter writer);

[C++] public: virtual void RenderBeginTag(HtmlTextWriter\* writer);

[VB] Overridable Public Sub RenderBeginTag(ByVal writer As HtmlTextWriter)

[JScript] public function RenderBeginTag(writer: HtmlTextWriter);

Description

Renders the HTML begin tag of the control into the specified writer. This method is primarily used by control developers.

This is made public so other controls can render multiple controls in between the begin tag and the end tag. The output stream that renders HTML content to the client.

RenderContents

[C#] protected virtual void RenderContents(HtmlTextWriter writer);

 $[C++]\ protected:\ virtual\ void\ RenderContents(HtmlTextWriter*\ writer);$ 

[VB] Overridable Protected Sub RenderContents(ByVal writer As

HtmlTextWriter)

[JScript] protected function RenderContents(writer: HtmlTextWriter);

25

## Description

1

2

3

4

5

6

8

10

11

12

13

14

15

16

17

19

20

21

22

23

24

25

Renders the contents of the control into the specified writer. This method is primarily used by control developers. The output stream that renders HTML content to the client.

RenderEndTag

[C#] public virtual void RenderEndTag(HtmlTextWriter writer);

[C++] public: virtual void RenderEndTag(HtmlTextWriter\* writer);

[VB] Overridable Public Sub RenderEndTag(ByVal writer As HtmlTextWriter)

[JScript] public function RenderEndTag(writer: HtmlTextWriter);

## Description

Renders the HTML end tag of the control into the specified writer. This method is primarily used by control developers.

This is made public so other controls can render multiple controls in between the begin tag and the end tag. The output stream that renders HTML content to the client.

SaveViewState

[C#] protected override object SaveViewState();

[C++] protected: Object\* SaveViewState();

[VB] Overrides Protected Function SaveViewState() As Object

[JScript] protected override function SaveViewState(): Object;

20

21

22

23

24

25

$\mathbf{r}$			, •	
1)	esc.	rin	tin	n
v	COC.	$\iota \nu \rho$	$\iota\iota\upsilon$	,,

1

2

3

5

A protected method. Saves any state that was modified after the System.Web.UI.WebControls.Style.TrackViewState method was invoked.

IAttributeAccessor.GetAttribute

[C#] string IAttributeAccessor.GetAttribute(string name);

[C++] String\* IAttributeAccessor::GetAttribute(String\* name);

[VB] Function GetAttribute(ByVal name As String) As String Implements

IAttributeAccessor.GetAttribute

[JScript] function IAttributeAccessor.GetAttribute(name : String) : String;

IAttributeAccessor.SetAttribute

[C#] void IAttributeAccessor.SetAttribute(string name, string value);

[C++] void IAttributeAccessor::SetAttribute(String\* name, String\* value);

[VB] Sub SetAttribute(ByVal name As String, ByVal value As String)

Implements IAttributeAccessor.SetAttribute

[JScript] function IAttributeAccessor.SetAttribute(name: String, value: String);

**TrackViewState** 

[C#] protected override void TrackViewState();

[C++] protected: void TrackViewState();

[VB] Overrides Protected Sub TrackViewState()

[JScript] protected override function TrackViewState();

lee **②**hayes pic 509+324+9256 2229 *MS1-863US.APP* 

1	
2	Description
3	Marks the beginning for tracking state changes on the control. Any changes
4	made after "mark" will be tracked and saved as part of the control viewstate.
5	Xml class (System.Web.UI.WebControls)
6	TrackViewState
7	
8	
9	Description
10	Displays the contents of an XML document or the results of an XSL
11	Transform.
12	Use the System.Web.UI.WebControls.Xml control to display the contents
13	of an XML document or the results of an XSL Transform.
14	Xml
15	Example Syntax:
16	TrackViewState
17	
18	[C#] public Xml();
19	[C++] public: Xml();
20	[VB] Public Sub New()
21	[JScript] public function Xml();
22	ChildControlsCreated
23	ClientID
24	Context
25	Controls
	3 4 5 6 7 8 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

1	Document
2	TrackViewState
3	
4	
5	Description
6	Gets or sets the <b>System.Xml.XmlDocument</b> object to display.
7	DocumentContent
	TrackViewState
8	Track view State
9	[C#] public string DocumentContent {get; set;}
10	[C++] public: property String* get_DocumentContent();public: property void
11	
12	set_DocumentContent(String*);
13	[VB] Public Property DocumentContent As String
14	[JScript] public function get DocumentContent(): String; public function set
15	DocumentContent(String);
16	
17	Description
18	Gets or sets a string that contains the XML document to display in the
19	System.Web.UI.WebControls.Xml control.
20	DocumentSource
21	TrackViewState
22	
23	[C#] public string DocumentSource {get; set;}
24	[C++] public:property String* get_DocumentSource();public:property void
25	set_DocumentSource(String*);

[VB] Public Property DocumentSource As String [JScript] public function get DocumentSource(): String; public function set DocumentSource(String); 3 Description Gets or sets the URL to an XML document to display in the 6 System.Web.UI.WebControls.Xml control. 7 **EnableViewState** 8 **Events** HasChildViewState 10 ID 11 IsTrackingViewState 12 NamingContainer 13 Page 14 Parent 15 Site 16 **TemplateSourceDirectory** Transform **TrackViewState** Description 22 Gets or sets the System.Xml.Xsl.XslTransform object that formats the 23 XML document before it is written to the output stream. 24 TransformArgumentList 25

	E ST	
i,		
÷		
i,		
;	ij,	
ľ,	ũ	
ŧ,	100	
:		
13		
ľ,	j	
	ų	
ļ	::lla	
ij	ij	
ť,		
ŀ	::ic	

1	TrackViewState
2	
3	[C#] public XsltArgumentList TransformArgumentList {get; set;}
4	[C++] public:property XsltArgumentList*
5	get_TransformArgumentList();public:property void
6	set_TransformArgumentList(XsltArgumentList*);
7	[VB] Public Property TransformArgumentList As XsltArgumentList
8	[JScript] public function get TransformArgumentList(): XsltArgumentList;public
9	function set TransformArgumentList(XsltArgumentList);
10	
11	Description
12	
13	TransformSource
14	TrackViewState
15	
16	[C#] public string TransformSource {get; set;}
17	[C++] public:property String* get_TransformSource();public:property void
18	set_TransformSource(String*);
19	[VB] Public Property TransformSource As String
20	[JScript] public function get TransformSource(): String; public function set
21	TransformSource(String);
22	
23	Description
24	Gets or sets the URL to an XSL Transform document that formats the XMI
25	document before it is written to the output stream.

	1	UniqueID
	2	ViewState
	3	ViewStateIgnoresCase
	4	Visible
	5	AddParsedSubObject
	6	
	7	[C#] protected override void AddParsedSubObject(object obj);
	8	[C++] protected: void AddParsedSubObject(Object* obj);
	9	[VB] Overrides Protected Sub AddParsedSubObject(ByVal obj As Object)
1	10	[JScript] protected override function AddParsedSubObject(obj : Object);
	11	
, , , , , , , , , , , , , , , , , , ,	12	Description
	13	
	14	Render
	15	
	16	[C#] protected override void Render(HtmlTextWriter output);
	17	[C++] protected: void Render(HtmlTextWriter* output);
	18	[VB] Overrides Protected Sub Render(ByVal output As HtmlTextWriter)
	19	[JScript] protected override function Render(output : HtmlTextWriter);
	20	
	21	Description
	22	Renders the results to the o
	23	
	24	
	25	

### EXEMPLARY COMPUTING SYSTEM AND ENVIRONMENT

Fig. 4 illustrates an example of a suitable computing environment 400 within which the programming framework 132 may be implemented (either fully or partially). The computing environment 400 may be utilized in the computer and network architectures described herein.

The exemplary computing environment 400 is only one example of a computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the computer and network architectures. Neither should the computing environment 400 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary computing environment 400.

The framework 132 may be implemented with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use include, but are not limited to, personal computers, server computers, multiprocessor systems, microprocessor-based systems, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and so on. Compact or subset versions of the framework may also be implemented in clients of limited resources, such as cellular phones, personal digital assistants, handheld computers, or other communication/computing devices.

The framework 132 may be described in the general context of computerexecutable instructions, such as program modules, being executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks

or implement particular abstract data types. The framework 132 may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

The computing environment 400 includes a general-purpose computing device in the form of a computer 402. The components of computer 402 can include, by are not limited to, one or more processors or processing units 404, a system memory 406, and a system bus 408 that couples various system components including the processor 404 to the system memory 406.

The system bus 408 represents one or more of several possible types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, such architectures can include an Industry Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA) local bus, and a Peripheral Component Interconnects (PCI) bus also known as a Mezzanine bus.

Computer 402 typically includes a variety of computer readable media. Such media can be any available media that is accessible by computer 402 and includes both volatile and non-volatile media, removable and non-removable media.

The system memory 406 includes computer readable media in the form of volatile memory, such as random access memory (RAM) 410, and/or non-volatile memory, such as read only memory (ROM) 412. A basic input/output system

lee@hayes pik 509+324+9256 2236

(BIOS) 414, containing the basic routines that help to transfer information between elements within computer 402, such as during start-up, is stored in ROM 412. RAM 410 typically contains data and/or program modules that are immediately accessible to and/or presently operated on by the processing unit 404.

Computer 402 may also include other removable/non-removable, volatile/non-volatile computer storage media. By way of example, Fig. 4 illustrates a hard disk drive 416 for reading from and writing to a non-removable, non-volatile magnetic media (not shown), a magnetic disk drive 418 for reading from and writing to a removable, non-volatile magnetic disk 420 (e.g., a "floppy disk"), and an optical disk drive 422 for reading from and/or writing to a removable, non-volatile optical disk 424 such as a CD-ROM, DVD-ROM, or other optical media. The hard disk drive 416, magnetic disk drive 418, and optical disk drive 422 are each connected to the system bus 408 by one or more data media interfaces 426. Alternatively, the hard disk drive 416, magnetic disk drive 418, and optical disk drive 422 can be connected to the system bus 408 by one or more interfaces (not shown).

The disk drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules, and other data for computer 402. Although the example illustrates a hard disk 416, a removable magnetic disk 420, and a removable optical disk 424, it is to be appreciated that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes or other magnetic storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or other optical storage, random access memories (RAM), read only memories (ROM), electrically erasable programmable read-only memory (EEPROM), and

the like, can also be utilized to implement the exemplary computing system and environment.

Any number of program modules can be stored on the hard disk 416, magnetic disk 420, optical disk 424, ROM 412, and/or RAM 410, including by way of example, an operating system 426, one or more application programs 428, other program modules 430, and program data 432. Each of the operating system 426, one or more application programs 428, other program modules 430, and program data 432 (or some combination thereof) may include elements of the programming framework 132.

A user can enter commands and information into computer 402 via input devices such as a keyboard 434 and a pointing device 436 (e.g., a "mouse"). Other input devices 438 (not shown specifically) may include a microphone, joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and other input devices are connected to the processing unit 404 via input/output interfaces 440 that are coupled to the system bus 408, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB).

A monitor 442 or other type of display device can also be connected to the system bus 408 via an interface, such as a video adapter 444. In addition to the monitor 442, other output peripheral devices can include components such as speakers (not shown) and a printer 446 which can be connected to computer 402 via the input/output interfaces 440.

Computer 402 can operate in a networked environment using logical connections to one or more remote computers, such as a remote computing device 448. By way of example, the remote computing device 448 can be a personal

computer, portable computer, a server, a router, a network computer, a peer device or other common network node, and so on. The remote computing device 448 is illustrated as a portable computer that can include many or all of the elements and features described herein relative to computer 402.

Logical connections between computer 402 and the remote computer 448 are depicted as a local area network (LAN) 450 and a general wide area network (WAN) 452. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When implemented in a LAN networking environment, the computer 402 is connected to a local network 450 via a network interface or adapter 454. When implemented in a WAN networking environment, the computer 402 typically includes a modem 456 or other means for establishing communications over the wide network 452. The modem 456, which can be internal or external to computer 402, can be connected to the system bus 408 via the input/output interfaces 440 or other appropriate mechanisms. It is to be appreciated that the illustrated network connections are exemplary and that other means of establishing communication link(s) between the computers 402 and 448 can be employed.

In a networked environment, such as that illustrated with computing environment 400, program modules depicted relative to the computer 402, or portions thereof, may be stored in a remote memory storage device. By way of example, remote application programs 458 reside on a memory device of remote computer 448. For purposes of illustration, application programs and other executable program components such as the operating system are illustrated herein as discrete blocks, although it is recognized that such programs and components

lee@hayes pilc 509-324-9256 2239 MS1-863US.APP

reside at various times in different storage components of the computing device 402, and are executed by the data processor(s) of the computer.

An implementation of the framework 132, and particularly, the API 142 or calls made to the API 142, may be stored on or transmitted across some form of computer readable media. Computer readable media can be any available media that can be accessed by a computer. By way of example, and not limitation, computer readable media may comprise "computer storage media" and "communications media." "Computer storage media" include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computer.

"Communication media" typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier wave or other transport mechanism. Communication media also includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and